Optimality

www.opt.reapress.com

Opt. Vol. 3, No. 2 (2026) 77–110.

Paper Type: Original Article

New Applications of Hyperstructures and Superhyperstructures: Queue, Markov Chains, Intervals, Logic, and Systems

Takaaki Fujita*

Independent Researcher, Tokyo, Japan; Takaaki.fujita060@gmail.com.

Citation:

Received: 19 September 2025 *Revised: 25 November 2025* *Accepted: 27 January 2026*	Fujita, T. (2026). New Applications of hyperstructures and superhyperstructures: Queue, markov chains, intervals, logic, and systems. *Optimality*, *3*(2), 77-110.

Abstract

Mathematical structures can be systematically extended to hyperstructures and superhyperstructures through the use of power sets and iterated power-set constructions. Such extensions provide a flexible framework for modeling hierarchical, multilevel, and set-valued phenomena across diverse mathematical and applied domains. Representative examples include superhypergraphs [1], superhyperalgebras, and superhyperfuzzy sets [2], which have recently attracted growing attention. Despite this progress, research on superhyperstructures is still at an early stage, and many potential applications, structural properties, and related concepts remain unexplored. Motivated by this gap, this paper investigates hyperstructural and superhyperstructural extensions of queues, Markov chains, intervals, logical systems, and discrete-time systems. For each case, we formally define the corresponding hyper and superhyper models, analyze their fundamental mathematical properties, and present illustrative examples that clarify their behavior and hierarchical nature. These results aim to broaden the theoretical foundation of superhyperstructures and stimulate further research into their applications and connections with existing mathematical frameworks.

Keywords: Hyperstructure, Superhyperstructure, Queue, Markov chains, Intervals, Logic, Systems.

1|Preliminaries

This section collects the basic notions used throughout the paper. Unless explicitly stated otherwise, all sets and structures are finite.

Corresponding Author: Takaaki.fujita060@gmail.com

https://doi.org/10.22105/opt.v3i1.111

1.1|Classical Structures, Hyperstructures, and n-Superhyperstructures

Hyperstructures and superhyperstructures are obtained by allowing algebraic operations to return sets, and then iterating this idea via iterated powersets [3]. We first fix notation for powersets and their iterations.

Definition 1 (Base set)**.** A *base set* is a nonempty finite set S. All higher-level objects in this paper (subsets, iterated subsets, hyperoutputs, and superhyperoutputs) are ultimately built from elements of S.

Definition 2 (Powerset and nonempty powerset) ([4])**.** Let S be a set. The *powerset* of S is

$$\mathcal{P}(S) := \{A \mid A \subseteq S\}.$$

The *nonempty powerset* of S is

$$\mathcal{P}^*(S) := \mathcal{P}(S) \setminus \{\emptyset\}.$$

Definition 3 (Iterated powerset and iterated nonempty powerset) ([3])**.** Let S be a set and let $n \in \mathbb{N}_0$. Define recursively

$$\mathcal{P}^{\,0}(S) := S, \qquad \mathcal{P}^{\,n+1}(S) := \mathcal{P}\big(\mathcal{P}^{\,n}(S)\big).$$

Similarly, define the iterated *nonempty* powerset by

$$(\mathcal{P}^*)^{\,0}(S) := S, \qquad (\mathcal{P}^*)^{\,n+1}(S) := \mathcal{P}^*\big((\mathcal{P}^*)^{\,n}(S)\big).$$

Definition 4 (Classical operation and classical structure [3]) · Let H be a nonempty set and let $m \geq 1$. A *classical m-ary operation* on H is a function

$$\# : H^m \to H.$$

A *classical structure* is a tuple

$$\mathbb{A} = (H, \#_1, \ldots, \#_t)$$

equipped with specified axioms relating the operations (e.g., associativity, distributivity, identities), depending on the intended algebraic type.

Definition 5 (Hyperoperation) ([5])**.** Let H be a nonempty set and let $m \geq 1$. A *hyperoperation* (of arity m) on H is a map

$$\circ : H^m \longrightarrow \mathcal{P}^*(H),$$

so the value $\circ(x_1, \ldots, x_m)$ is a *nonempty subset* of H rather than a single element.

Definition 6 (Hyperstructure) [3]. A *hyperstructure* is a tuple

$$\mathbb{H} = (H, \circ_1, \ldots, \circ_t)$$

where H is a nonempty set and each $\circ_j$ is a hyperoperation on H (possibly of its own arity), together with axioms appropriate to the model under consideration.

Definition 7 (Superhyperoperation of order (m, n)) ([3])**.** Let H be a nonempty set, and fix integers $m \geq 1$ and $n \geq 1$. A *(classical-type) (m, n)-superhyperoperation* on H is a map

$$\odot^{(m,n)} : H^m \longrightarrow (\mathcal{P}^*)^{\,n}(H),$$

so each output is a nonempty element of the n-fold iterated powerset. If one allows the empty set at some level, one may instead take codomain $\mathcal{P}^{\,n}(H)$; we refer to this as the *Neutrosophic-type* variant.

Definition 8 (*n*-superhyperstructure) ([3]). Fix $n \geq 1$. An *n-superhyperstructure* is a tuple

$$\mathbb{SH}^{(n)} = (H, \odot_1, \ldots, \odot_t)$$

where H is a nonempty set and each $\odot_j$ is a superhyperoperation

$$\odot_j : \ H^{m_j} \to (\mathcal{P}^*)^n(H)$$

for some arity $m_j \geq 1$, together with axioms suited to the application.

Example 1 (Group-based 2-superhyperstructure). Let $G = \mathbb{Z}_3 = \{0, 1, 2\}$ with addition modulo 3. Then

$$\mathcal{P}^1(G) = \mathcal{P}(G), \qquad \mathcal{P}^2(G) = \mathcal{P}\big(\mathcal{P}(G)\big).$$

Define a binary operation

$$\star : \ \mathcal{P}^2(G) \times \mathcal{P}^2(G) \longrightarrow \mathcal{P}^2(G)$$

by

$$A \star B := \Big\{ X \oplus Y \ \Big| \ X \in A, \ Y \in B \Big\}, \qquad X \oplus Y := \{ x + y \bmod 3 \mid x \in X, \ y \in Y \}.$$

For instance, if

$$A = \big\{\{0, 1\}, \{2\}\big\}, \qquad B = \big\{\{1\}\big\},$$

then

$$\{0, 1\} \oplus \{1\} = \{1, 2\}, \qquad \{2\} \oplus \{1\} = \{0\},$$

so

$$A \star B = \big\{\{1, 2\}, \{0\}\big\} \in \mathcal{P}^2(G).$$

Thus $(\mathcal{P}^2(G), \star)$ is a concrete example of a 2-level superhyperstructure built from the group $(G, +)$.

Remark 1. The same lifting principle extends to any level $n \geq 2$: one replaces $\mathcal{P}^2(H)$ by $\mathcal{P}^n(H)$ (or $(\mathcal{P}^*)^n(H)$) and defines the lifted operation by taking pointwise images of lower-level combinations.

For reference, a comparison of Classical Structures, HyperStructures, and n-SuperHyperStructures is presented in Table 1. It is expected that n-SuperHyperStructures offer advantages such as clearly modeling hierarchical and multi-level real-world concepts.

Notion	Carrier (Universe)	Operation Type	Codomain of an m-ary Operation
Classical Structure	H	classical operation	$\# : \ H^m \to H$
HyperStructure	H	hyperoperation	$\circ : \ H^m \to \mathcal{P}^*(H)$
SuperHyperStructure (level n)	H	(m, n)-superhyperoperation	$\odot^{(m,n)} : \ H^m \to (\mathcal{P}^*)^n(H)$

TABLE 1. A compact comparison of Classical Structures, HyperStructures, and n-SuperHyperStructures. Here $\mathcal{P}^*(H) = \mathcal{P}(H) \setminus \{\emptyset\}$ and $(\mathcal{P}^*)^n(H)$ is the n-fold iterated nonempty powerset.

2|New Concepts using SuperhyperStructure

This section presents new concepts constructed using Superhyperstructures.

2.1|Hyperqueue and Superhyperqueue

A queue is a linear data structure in which elements are inserted at the rear and removed from the front, following the FIFO principle [6, 7, 8]. In this subsection, we extend this concept to hyperqueues and superhyperqueues.

Definition 9 (Queue). Let E be a nonempty set of elements. A *queue* over E is an abstract data type defined by:

$$Q = E^*$$

the set of all finite sequences (lists) of elements of E. The following operations are defined on Q:

- $empty : Q$, the empty queue:
$$empty = \langle\,\rangle.$$
- $enqueue : Q \times E \to Q$, which adds an element at the tail:
$$enqueue(q, e) = q \,\|\, \langle e \rangle,$$
where $\|$ denotes sequence concatenation.
- $dequeue : Q \to Q \times E$, which removes and returns the head element:
$$dequeue(q) = \begin{cases} (q',\, e), & \text{if } q = \langle e \rangle \,\|\, q', \\ \text{undefined}, & \text{if } q = \langle\,\rangle. \end{cases}$$
- $isEmpty : Q \to \{true, false\}$,
$$isEmpty(q) = \begin{cases} true, & q = \langle\,\rangle, \\ false, & q \neq \langle\,\rangle. \end{cases}$$

These operations satisfy the following axioms for all $q \in Q$ and $e \in E$:

(1) $isEmpty(empty) = true$.

(2) $isEmpty(enqueue(q, e)) = false$.

(3) If $dequeue(q) = (q', e)$, then
$$q = \langle e \rangle \,\|\, q', \quad enqueue(q', e) \neq q' \,\|\, e$$
and repeated dequeue operations remove elements in the same order they were enqueued (FIFO property).

Example 2 (Customer support ticket queue)**.** A Customer Support Ticket Queue is a system that manages and orders customer service requests for processing, typically in a FIFO manner (cf.[9, 10, 11]). Let the set of customer support tickets be

$$E = \{\mathit{T1},\ \mathit{T2},\ \mathit{T3}\}.$$

We illustrate a sequence of queue operations on $Q = E^*$.

Initial state:

$$Q_0 = empty = \langle\,\rangle, \qquad isEmpty(Q_0) = true.$$

Enqueue T1:

$$Q_1 = enqueue(Q_0, \mathit{T1}) = \langle \mathit{T1} \rangle, \qquad isEmpty(Q_1) = false.$$

Enqueue T2:

$$Q_2 = enqueue(Q_1, \mathit{T2}) = \langle \mathit{T1},\ \mathit{T2} \rangle.$$

Enqueue T3:

$$Q_3 = enqueue(Q_2, \mathit{T3}) = \langle \mathit{T1},\ \mathit{T2},\ \mathit{T3} \rangle.$$

Dequeue twice:

$$dequeue(Q_3) = (Q_2',\ T1), \quad Q_2' = \langle T2,\ T3\rangle,$$
$$dequeue(Q_2') = (Q_1',\ T2), \quad Q_1' = \langle T3\rangle.$$

Check remaining and empty:

$$isEmpty(Q_1') = false, \quad dequeue(Q_1') = (Q_0,\ T3), \quad isEmpty(Q_0) = true.$$

Summary of FIFO behavior:

- Tickets $T1, T2, T3$ were enqueued in that order.
- They were dequeued in the same order $T1 \to T2 \to T3$.
- At each step, *isEmpty* correctly reports whether any tickets remain.

This concrete example demonstrates the FIFO discipline and all queue operations in detail.

The definition of a hyperqueue is given below.

Definition 10 (Hyperqueue). Let E be a nonempty set and let $Q = E^*$ be the classical queue over E. Define the *hyperqueue* on E as the hyperstructure

$$\mathrm{HQ} \ = \ \big(P(Q),\ \odot_{\mathrm{enq}},\ \odot_{\mathrm{deq}}\big),$$

where $P(Q)$ is the powerset of Q, and the operations

$$\odot_{\mathrm{enq}} : P(Q) \times E \ \longrightarrow \ P(Q), \qquad \odot_{\mathrm{deq}} : P(Q) \ \longrightarrow \ P(Q \times E)$$

are given by

$$\odot_{\mathrm{enq}} \ e = \big\{\ enqueue(q, e) \mid q \in A\big\}, \quad A \odot_{\mathrm{deq}} = \big\{(q', e) \mid q \in A,\ (q', e) = dequeue(q)\big\}.$$

Remark 2. By construction, $\odot_{\mathrm{enq}}$ and $\odot_{\mathrm{deq}}$ are hyperoperations in the sense that their codomains are powersets of the classical domains. Hence HQ is a hyperstructure on the base set Q.

Theorem 1. *The classical queue* $(Q, enqueue, dequeue)$ *is recovered from* HQ *by restricting to singleton subsets. Concretely, for every* $q \in Q$ *and* $e \in E$,

$$\{q\} \odot_{\mathrm{enq}} e = \big\{enqueue(q, e)\big\}, \quad \{q\}\odot_{\mathrm{deq}} = \big\{dequeue(q)\big\}.$$

Proof: Recall that the hyperqueue $\mathrm{HQ} = (P(Q), \odot_{\mathrm{enq}}, \odot_{\mathrm{deq}})$ is defined by

$$A \ \odot_{\mathrm{enq}} \ e = \big\{enqueue(q, e) \mid q \in A\big\}, \quad A \odot_{\mathrm{deq}} = \big\{(q', e) \mid q \in A,\ (q', e) = dequeue(q)\big\}.$$

We must show that when A is the singleton $\{q\}$, these hyperoperations collapse exactly to the classical operations on q.

(i) Enqueue. Let $q \in Q$ and $e \in E$. Then by definition

$$\{q\} \ \odot_{\mathrm{enq}} \ e = \big\{enqueue(x, e) \mid x \in \{q\}\big\}.$$

Since the only element of $\{q\}$ is q itself, the comprehension reduces to

$$\{q\} \ \odot_{\mathrm{enq}} \ e = \{enqueue(q, e)\}.$$

Hence every element of the left-hand side is of the form $enqueue(q, e)$, and clearly $enqueue(q, e)$ lies in the set on the right. This establishes the equality

$$\{q\} \odot_{\mathrm{enq}} e = \{enqueue(q, e)\}.$$

(ii) Dequeue. Similarly, by definition,

$$\{q\} \odot_{\mathrm{deq}} = \big\{(x', f) \mid x \in \{q\},\ (x', f) = dequeue(x)\big\}.$$

Again the only choice for x is q. Therefore

$$\{q\} \odot_{\text{deq}} = \{\, \mathit{dequeue}(q) \,\},$$

where $\mathit{dequeue}(q)$ denotes the unique pair (q', f) returned by the classical dequeue on q. Thus

$$\{q\} \odot_{\text{deq}} = \{\mathit{dequeue}(q)\}.$$

Since both hyperoperations agree on singletons with the classical enqueue and dequeue, we conclude that the entire classical queue $(Q, \mathit{enqueue}, \mathit{dequeue})$ is recovered by restricting HQ to singleton subsets.

Example 3 (Customer Ticket Hyperqueue)**.** Let the set of ticket identifiers be

$$E = \{\mathit{T1},\ \mathit{T2}\}, \qquad Q = E^* = \{\langle\rangle,\ \langle \mathit{T1}\rangle,\ \langle \mathit{T2}\rangle,\ \langle \mathit{T1}, \mathit{T2}\rangle,\ \dots\}.$$

Consider the subset of queues

$$A \ = \ \big\{\, \langle \mathit{T1}\rangle,\ \langle \mathit{T2},\ \mathit{T1}\rangle \big\} \ \in \ P(Q).$$

Hyperenqueue Enqueueing ticket $\mathit{T2}$ on every queue in A gives

$$A \ \odot_{\text{enq}} \ \mathit{T2} = \big\{\mathit{enqueue}(\langle \mathit{T1}\rangle, \mathit{T2}),\ \mathit{enqueue}(\langle \mathit{T2}, \mathit{T1}\rangle, \mathit{T2})\big\} = \big\{\langle \mathit{T1}, \mathit{T2}\rangle, \langle \mathit{T2}, \mathit{T1}, \mathit{T2}\rangle\big\}.$$

Hyperdequeue Removing the head ticket from each queue in A yields pairs of (remaining-queue, dequeued-ticket):

$$A \odot_{\text{deq}} = \big\{(q', e) \ \big|\ q \in A,\ (q', e) = \mathit{dequeue}(q)\big\} = \big\{(\langle\rangle, \mathit{T1}),\ (\langle \mathit{T1}\rangle, \mathit{T2})\big\}.$$

Iterated Hyperoperations One can combine hyperenqueue and hyperdequeue on sets of queues. For example, first enqueue $\mathit{T2}$ then dequeue:

$$\big(A \odot_{\text{enq}} \mathit{T2}\big) \odot_{\text{deq}} = \big\{(\langle \mathit{T2}\rangle, \mathit{T1}),\ (\langle \mathit{T1}, \mathit{T2}\rangle, \mathit{T2})\big\}.$$

This demonstrates how $\text{HQ} = (P(Q), \odot_{\text{enq}}, \odot_{\text{deq}})$ systematically generalizes classical queue operations to sets of queues.

The definition of a n-SuperHyperqueue is given below.

Definition 11 (n-SuperHyperqueue)**.** Let E be a nonempty set, and let $Q = E^*$ be the classical queue over E. For each integer $n \geq 0$, define the *n-SuperHyperqueue*

$$\text{SNHQ}_n = \big(\mathcal{P}_n(Q),\ \odot_{\text{enq}}^{(n)},\ \odot_{\text{deq}}^{(n)}\big),$$

where the two superhyperoperations are given by:

$$\begin{gathered}
\odot_{\text{enq}}^{(n)} : \mathcal{P}_n(Q) \times E \ \longrightarrow \ \mathcal{P}_n(Q), \\
\odot_{\text{deq}}^{(n)} : \mathcal{P}_n(Q) \ \longrightarrow \ \mathcal{P}_n\big(Q \times E\big), \\
A \ \odot_{\text{enq}}^{(n)} \ e = \big\{\, X \ \odot_{\text{enq}}^{(n-1)} \ e \ \big|\ X \in A\big\}, \\
A \odot_{\text{deq}}^{(n)} = \big\{\, X \odot_{\text{deq}}^{(n-1)} \ \big|\ X \in A\big\},
\end{gathered}$$

with the *base case* operations $\odot_{\text{enq}}^{(0)} = \mathit{enqueue}$ and $\odot_{\text{deq}}^{(0)} = \mathit{dequeue}$.

Remark 3. By construction, each $\odot^{(n)}$ takes its inputs in $\mathcal{P}_n(Q)$ (and E for enqueue) and returns values in $\mathcal{P}_n(Q)$ or $\mathcal{P}_n(Q \times E)$. Hence SNHQ_n is an (m, n)-Superhyperstructure with $m = 2$.

Proposition 2.9. SNHQ_n *indeed carries the structure of an n-Superhyperstructure on the base set Q.*

Proof: By Definition 2.7, both operations $\odot_{\text{enq}}^{(n)}$ and $\odot_{\text{deq}}^{(n)}$ map into the n-th powerset $\mathcal{P}_n(\,\cdot\,)$. This exactly matches the requirement for an (m, n)-superhyperoperation (here $m = 2$), so SNHQ_n is an n-superhyperstructure.

Theorem 2. *For $n = 0$ and $n = 1$, the n-SuperHyperqueue specializes to the classical queue and the Hyperqueue, respectively:*

$$\mathrm{SNHQ}_0 = \big(Q, \mathit{enqueue}, \mathit{dequeue}\big), \qquad \mathrm{SNHQ}_1 = \big(P(Q), \odot_{\mathrm{enq}}, \odot_{\mathrm{deq}}\big) = \mathrm{HQ}.$$

Proof: **Case** $n = 0$**:** By definition $\mathcal{P}_0(Q) = Q$, and $\odot_{\mathrm{enq}}^{(0)} = \mathit{enqueue}$, $\odot_{\mathrm{deq}}^{(0)} = \mathit{dequeue}$. Thus SNHQ_0 is exactly the classical queue.

Case $n = 1$**:** We have $\mathcal{P}_1(Q) = P(Q)$. By the recursion

$$A \odot_{\mathrm{enq}}^{(1)} e = \{\, X \odot_{\mathrm{enq}}^{(0)} e \mid X \in A\} = \{\mathit{enqueue}(q, e) \mid q \in A\},$$

and similarly for $\odot_{\mathrm{deq}}^{(1)}$. These coincide with the hyperqueue operations $\odot_{\mathrm{enq}}, \odot_{\mathrm{deq}}$. Therefore $\mathrm{SNHQ}_1 = \mathrm{HQ}$.

Theorem 3 (Closure under Enqueue and Dequeue)**.** *For every integer $n \geq 0$, every $e \in E$, and every $A \in \mathcal{P}_n(Q)$:*

$$A \ \odot_{\mathrm{enq}}^{(n)} \ e \ \in \ \mathcal{P}_n(Q), \qquad A \odot_{\mathrm{deq}}^{(n)} \ \in \ \mathcal{P}_n(Q \times E).$$

Proof: We prove both statements by induction on n.

Base case ($n = 0$). Here $\mathcal{P}_0(Q) = Q$, $\odot_{\mathrm{enq}}^{(0)} = \mathit{enqueue}$, and $\odot_{\mathrm{deq}}^{(0)} = \mathit{dequeue}$. Clearly $\mathit{enqueue}(q, e) \in Q$ and $\mathit{dequeue}(q) \in Q \times E$ for every $q \in Q$, so the closure holds.

Inductive step. Assume the claim holds for $n - 1$. Let $A \in \mathcal{P}_n(Q) = \mathcal{P}(\mathcal{P}_{n-1}(Q))$. Then by definition

$$A \odot_{\mathrm{enq}}^{(n)} e = \{\, X \odot_{\mathrm{enq}}^{(n-1)} e \mid X \in A\}.$$

Since each $X \in \mathcal{P}_{n-1}(Q)$, the inductive hypothesis ensures $X \odot_{\mathrm{enq}}^{(n-1)} e \in \mathcal{P}_{n-1}(Q)$. Hence the entire set lies in $\mathcal{P}\big(\mathcal{P}_{n-1}(Q)\big) = \mathcal{P}_n(Q)$. The argument for $\odot_{\mathrm{deq}}^{(n)}$ is identical, replacing enq by deq and Q by $Q \times E$.

Theorem 4 (Restriction to Singletons)**.** *For any $n \geq 1$, any $X \in \mathcal{P}_{n-1}(Q)$, and any $e \in E$,*

$$\{X\} \ \odot_{\mathrm{enq}}^{(n)} \ e = \big\{\, X \ \odot_{\mathrm{enq}}^{(n-1)} \ e\big\}, \quad \{X\} \odot_{\mathrm{deq}}^{(n)} = \big\{\, X \odot_{\mathrm{deq}}^{(n-1)}\big\}.$$

Consequently, the level-n superhyperqueue restricted to singleton subsets recovers the level-$(n-1)$ superhyperqueue.

Proof: Since $\{X\} \subseteq \mathcal{P}_{n-1}(Q)$, by definition

$$\{X\} \odot_{\mathrm{enq}}^{(n)} e = \{\, Y \odot_{\mathrm{enq}}^{(n-1)} e \mid Y \in \{X\}\} = \{\, X \odot_{\mathrm{enq}}^{(n-1)} e\}.$$

The same reasoning applies to $\odot_{\mathrm{deq}}^{(n)}$. Hence every operation at level n on a singleton set exactly reproduces the corresponding level $(n - 1)$ operation on its unique element.

Example 4 (A 2-SuperHyperqueue)**.** Let

$$E = \{\mathit{T1},\ \mathit{T2}\}, \qquad Q = E^* = \{\langle\,\rangle,\ \langle \mathit{T1}\rangle,\ \langle \mathit{T2}\rangle,\ \langle \mathit{T1}, \mathit{T2}\rangle,\ \langle \mathit{T2}, \mathit{T1}\rangle,\ \ldots\}.$$

Then $\mathcal{P}_1(Q) = \mathcal{P}(Q)$ and $\mathcal{P}_2(Q) = \mathcal{P}(\mathcal{P}(Q))$. Choose the single second-level element

$$A \ = \ \big\{\{\,\langle \mathit{T1}\rangle,\ \langle \mathit{T2}, \mathit{T1}\rangle\}\big\} \ \in \ \mathcal{P}_2(Q),$$

where

$$X = \{\,\langle \mathit{T1}\rangle,\ \langle \mathit{T2}, \mathit{T1}\rangle\} \ \in \ \mathcal{P}_1(Q).$$

Second-Level Hyperenqueue. For $e = \mathit{T2}$,

$$\begin{aligned} A \ \odot_{\mathrm{enq}}^{(2)} \ \mathit{T2} &= \big\{\, X \odot_{\mathrm{enq}}^{(1)} \ \mathit{T2}\big\} = \Big\{\{\ \mathit{enqueue}(q, \mathit{T2}) \mid q \in X\}\Big\} \\ &= \Big\{\{\ \langle \mathit{T1}, \mathit{T2}\rangle,\ \langle \mathit{T2}, \mathit{T1}, \mathit{T2}\rangle\}\Big\}. \end{aligned}$$

Second-Level Hyperdequeue.

$$A \odot^{(2)}_{\text{deq}} = \{ X \odot^{(1)}_{\text{deq}} \} = \Big\{ \{ (q', e) \mid q \in X,\ (q', e) = \mathit{dequeue}(q) \} \Big\}$$
$$= \Big\{ \{ (\langle\rangle, \mathit{T1}),\ (\langle \mathit{T1} \rangle, \mathit{T2}) \} \Big\}.$$

Thus SNHQ_2 encodes two nested levels of queue-like operations:

- Level 1 ($\odot^{(1)}$) is the hyperqueue on sets of queues,
- Level 2 ($\odot^{(2)}$) lifts that to sets of sets of queues.

Each superhyperoperation systematically generalizes the classical queue operations across two hierarchical layers.

Example 5 (A real-world 3-superhyperqueue: Multi-Tier task pipelines). A task pipeline is a sequence of processing stages where tasks flow step-by-step, each stage performing a specific operation. Suppose a company processes support tickets through three hierarchical tiers: *Agent Teams* → *Departments* → *Divisions*. Let the basic ticket types be

$$E = \{ \mathit{T1},\ \mathit{T2} \}, \quad Q = E^*,$$

and form the iterated powersets

$$\mathcal{P}_1(Q) = \mathcal{P}(Q), \quad \mathcal{P}_2(Q) = \mathcal{P}(\mathcal{P}(Q)), \quad \mathcal{P}_3(Q) = \mathcal{P}(\mathcal{P}_2(Q)).$$

Level-1 (Agent Team) Queues:

$$X_1 = \{ \langle \mathit{T1} \rangle \}, \quad X_2 = \{ \langle \mathit{T2}, \mathit{T1} \rangle \} \ \in\ \mathcal{P}_1(Q).$$

Level-2 (Department) Collections:

$$U = \{ X_1,\ X_2 \} \ \in\ \mathcal{P}_2(Q).$$

Level-3 (Division) Supercollection:

$$A = \{ U \} \ \in\ \mathcal{P}_3(Q).$$

Third-Level Hyperenqueue. To enqueue a new ticket *T2* at the division level:

$$A \ \odot^{(3)}_{\text{enq}}\ \mathit{T2} = \{ U \odot^{(2)}_{\text{enq}}\ \mathit{T2} \}$$
$$= \Big\{ \{ X \odot^{(1)}_{\text{enq}}\ \mathit{T2} \mid X \in U \} \Big\}$$
$$= \Big\{ \{ \langle \mathit{T1}, \mathit{T2} \rangle,\ \langle \mathit{T2}, \mathit{T1}, \mathit{T2} \rangle \} \Big\}.$$

Third-Level Hyperdequeue. To dequeue at the division level:

$$A \odot^{(3)}_{\text{deq}} = \{ U \odot^{(2)}_{\text{deq}} \}$$
$$= \Big\{ \{ (q', e) \mid q \in U,\ (q', e) = \mathit{dequeue}(q) \} \Big\}$$
$$= \Big\{ \{ (\langle\rangle, \mathit{T1}),\ (\langle \mathit{T1} \rangle, \mathit{T2}) \} \Big\}.$$

Interpretation:

- *Level 1* ($\mathcal{P}_1$) are the queues handled by individual agent teams.
- *Level 2* groups those team-queues into department-wide collections.
- *Level 3* collects departments into a division-wide superqueue.
- A single division-level hyperenqueue/dequeue simultaneously enqueues/dequeues on every department's queues.

This illustrates how a 3-SuperHyperqueue models three hierarchical tiers of queue processing in a real-world organization.

For reference, Table 2 presents an overview of queues and their hyper and superhyper extensions.

Notion	Carrier (Universe)	Enqueue Output	Dequeue Output
Queue	$Q := E^*$ (finite sequences over E)	$enqueue : Q \times E \to Q$ (single next queue)	$dequeue : Q \to Q \times E$ (single pair, undefined on $\langle\rangle$)
Hyperqueue	$\mathcal{P}(Q)$ (sets of queues)	$\odot_{\text{enq}} : \mathcal{P}(Q) \times E \to \mathcal{P}(Q)$, $A \odot_{\text{enq}} e = \{enqueue(q, e) \mid q \in A\}$	$\odot_{\text{deq}} : \mathcal{P}(Q) \to \mathcal{P}(Q \times E)$, $A\odot_{\text{deq}} = \{(q', e) \mid q \in A,\ (q', e) = dequeue(q)\}$
n-SuperHyperqueue	$\mathcal{P}_n(Q)$ (iterated powerset, $n \geq 0$)	$\odot_{\text{enq}}^{(n)} : \mathcal{P}_n(Q) \times E \to \mathcal{P}_n(Q)$, $A \odot_{\text{enq}}^{(n)} e = \{X \odot_{\text{enq}}^{(n-1)} e \mid X \in A\}$	$\odot_{\text{deq}}^{(n)} : \mathcal{P}_n(Q) \to \mathcal{P}_n(Q \times E)$, $A\odot_{\text{deq}}^{(n)} = \{X\odot_{\text{deq}}^{(n-1)} \mid X \in A\}$

TABLE 2. A compact overview of queues and their hyper/superhyper extensions. Here E is a nonempty set, $Q = E^*$, and $\mathcal{P}_0(Q) = Q$, $\mathcal{P}_{n+1}(Q) = \mathcal{P}(\mathcal{P}_n(Q))$. Special cases: $\text{SNHQ}_0 = (Q, enqueue, dequeue)$ and $\text{SNHQ}_1 = (\mathcal{P}(Q), \odot_{\text{enq}}, \odot_{\text{deq}})$.

2.2|Markov HyperChain and Markov SuperHyperChain

A Markov chain is a stochastic process in which the next state depends only on the current state and not on the prior history [12, 13, 14]. In this subsection, we extend this concept to Markov HyperChains and Markov SuperHyperChains and investigate their properties.

Definition 12 (Markov Chain). [12, 13] Let $(\Omega, \mathcal{F}, P)$ be a probability space and let S be a countable (or finite) set called the *state space*. A *discrete-time stochastic process* $\{X_n\}_{n=0}^{\infty}$, where each

$$X_n : \Omega \longrightarrow S,$$

is called a *(discrete-time) Markov chain* if for every $n \geq 0$ and every sequence of states $i_0, i_1, \ldots, i_{n+1} \in S$,

Definition 13 (Time-Homogeneity and Transition Matrix). A Markov chain $\{X_n\}$ is *time-homogeneous* if there exists a function

$$P : S \times S \longrightarrow [0, 1], \quad P(i, j) = \mathbb{P}(X_{n+1} = j \mid X_n = i),$$

independent of n. The array $\big(P(i, j)\big)_{i,j\in S}$ is called the *transition matrix*.

Definition 14 (Initial Distribution and n-Step Probabilities). The *initial distribution* $\pi^{(0)}$ is given by

$$\pi_i^{(0)} = \mathbb{P}(X_0 = i), \quad i \in S.$$

The *n-step transition probability* is

$$P^{(n)}(i, j) = \mathbb{P}\big(X_n = j \mid X_0 = i\big), \quad i, j \in S,\ n \geq 0.$$

Theorem 15 (Chapman–Kolmogorov Equations). [15] *For all $m, n \geq 0$ and $i, j \in S$,*

$$P^{(m+n)}(i, j) \ = \ \sum_{k\in S} P^{(m)}(i, k)\,\big[P^{(n)}(k, j)\big].$$

In matrix form, if P is the one-step transition matrix, then

$$P^{(n)} = P^n.$$

Remark 4. A state j is *absorbing* if $P(j,j) = 1$. A chain is *irreducible* if every state is reachable from every other. Additional classifications (recurrence/transience, periodicity, stationary distributions) can be developed from these definitions.

Example 6 (Weather Prediction Model). Weather prediction estimates future atmospheric conditions like temperature, rainfall, and wind using current data and numerical simulation models (cf.[16, 17, 18, 19]). A classical real-world application of a Markov chain is weather modeling, with a finite state space

$$S = \{\text{Sunny},\ \text{Cloudy},\ \text{Rainy}\}.$$

We assume a discrete daily process $\{X_n\}_{n=0}^{\infty}$, where X_n is the weather on day n.
Initial Distribution Suppose on day 0 the probabilities of each weather are

$$\pi^{(0)} = \big(\pi^{(0)}_{\text{Sunny}},\ \pi^{(0)}_{\text{Cloudy}},\ \pi^{(0)}_{\text{Rainy}}\big) = (0.50,\ 0.30,\ 0.20).$$

Transition Matrix We posit a time-homogeneous transition matrix P capturing the probabilities of tomorrow's weather given today's:

$$P = \big(P(i,j)\big)_{i,j\in S} = \begin{pmatrix} 0.80 & 0.15 & 0.05 \\ 0.20 & 0.60 & 0.20 \\ 0.20 & 0.30 & 0.50 \end{pmatrix},$$

where, for example, $P(\text{Sunny}, \text{Cloudy}) = 0.15$ and $P(\text{Rainy}, \text{Rainy}) = 0.50$.
One-Step Evolution The distribution on day 1 is

$$\pi^{(1)} \ = \ \pi^{(0)} P \ = \ (0.50, 0.30, 0.20) \begin{pmatrix} 0.80 & 0.15 & 0.05 \\ 0.20 & 0.60 & 0.20 \\ 0.20 & 0.30 & 0.50 \end{pmatrix} = (0.59,\ 0.345,\ 0.065).$$

Thus there is a 59% chance of Sunshine on day 1, 34.5% Cloudy, and 6.5% Rainy.
Two-Step Probabilities By the Chapman–Kolmogorov equation,

$$P^{(2)} \ = \ P^2,$$

so, for instance,

$$P^{(2)}(\text{Sunny}, \text{Rainy}) = \sum_{k\in S} P(\text{Sunny}, k)\, P(k, \text{Rainy}) = 0.8 \cdot 0.05 + 0.15 \cdot 0.20 + 0.05 \cdot 0.50 = 0.095.$$

Hence if today is Sunny, the chance of Rain tomorrow (two days ahead) is 9.5%.
Interpretation This simple weather-prediction Markov chain illustrates how one can:

- Encode the "memoryless" Markov property: tomorrow's weather depends only on today's.
- Use an initial distribution and transition matrix to compute future distributions.
- Apply Chapman–Kolmogorov to obtain multi-step forecasts.

The definition of a Markov HyperChain is given below.

Definition 16 (Markov HyperChain). Let S be a nonempty countable (or finite) set, called the *state space*. A *Markov HyperChain* on S is a pair

$$\text{MHC} \ = \ \big(S,\ H\big),$$

where

$$H \ : \ S \times S \ \longrightarrow \ \mathcal{P}([0,1]), \qquad (i,j) \ \mapsto \ H_{ij},$$

is called the *hypertransition matrix*, subject to:

(i) **Nonemptiness:**

$$H_{ij} \ \neq \ \varnothing, \quad \forall\, i,j \in S.$$

(ii) **Hypernormalization:** For each source state $i \in S$, there exists at least one *selection function* $p_i : S \to [0,1]$ such that

$$p_i(j) \in H_{ij} \quad \text{for every } j \in S, \quad \text{and} \quad \sum_{j \in S} p_i(j) \;=\; 1.$$

Remark 5. Because $H_{ij} \subseteq [0,1]$, the map H takes values in the powerset $\mathcal{P}([0,1])$. Thus H is a *hyperoperation* in the sense of hyperstructure theory, and MHC forms a *hyperstructure* on S.

Theorem 5 (Recovery of Classical Markov Chain)**.** *If for all $i, j \in S$ the hypertransition sets are singletons,*

$$H_{ij} \;=\; \big\{\, P(i,j) \big\},$$

then MHC $= (S, H)$ *is equivalent to the usual time-homogeneous Markov chain with transition matrix* $\big(P(i,j)\big)$.

Proof: Define $p_i(j) = P(i,j)$. Since $H_{ij} = \{P(i,j)\}$, we have $p_i(j) \in H_{ij}$ for each i, j. By the usual stochastic assumption, $\sum_j P(i,j) = 1$. Hence all axioms (i)–(ii) of the Markov HyperChain are satisfied, and no additional choices arise because each H_{ij} contains exactly one element. Thus the hyperstructure MHC reduces precisely to the classical Markov chain.

Theorem 6 (Hyperstructure Property)**.** *The mapping*

$$H : S \times S \;\longrightarrow\; \mathcal{P}([0,1])$$

is a binary hyperoperation*; therefore* MHC $= (S, H)$ *is a hyperstructure in the sense that its operation takes values in a powerset.*

Proof: By definition, for any $(i,j) \in S \times S$, one has $H_{ij} \subseteq [0,1]$. Hence

$$H(i,j) \;\in\; \mathcal{P}([0,1]),$$

which is exactly the requirement for a hyperoperation on S. No further conditions are needed beyond (i)–(ii), so MHC is a legitimate hyperstructure.

Example 7 (Three-State Markov HyperChain: Machine Condition Model)**.** Let

$$S = \{\, G,\, D,\, F\}$$

be the three-state space representing the health of a machine:

$$G = \text{Good}, \quad D = \text{Degraded}, \quad F = \text{Failed}.$$

We specify a *hypertransition matrix* $H : S \times S \to \mathcal{P}([0,1])$ by giving, for each $i \in S$, nonempty sets

$$H_{ij} \subseteq [0,1], \quad j \in S,$$

as follows:

	$j = G$	$j = D$	$j = F$
$i = G$	$\{0.7, 0.8\}$	$\{0.2, 0.25\}$	$\{0.0, 0.05\}$
$i = D$	$\{0.1, 0.15\}$	$\{0.6, 0.7\}$	$\{0.2, 0.3\}$
$i = F$	$\{0.05, 0.1\}$	$\{0.1, 0.2\}$	$\{0.7, 0.85\}$

Hypernormalization. For each source state i one must pick probabilities $p_i(j) \in H_{ij}$ so that $\sum_j p_i(j) = 1$. Two concrete selections are:

$$\text{(Selection 1)} \quad p_G^{(1)} = (0.8,\ 0.2,\ 0.0), \quad p_D^{(1)} = (0.15,\ 0.6,\ 0.25), \quad p_F^{(1)} = (0.10,\ 0.20,\ 0.70),$$

$$\text{(Selection 2)} \quad p_G^{(2)} = (0.7,\ 0.25,\ 0.05), \quad p_D^{(2)} = (0.10,\ 0.70,\ 0.20), \quad p_F^{(2)} = (0.05,\ 0.10,\ 0.85).$$

Induced Classical Chains. Each selection yields a classical transition matrix. For Selection 1:

$$P^{(1)} = \left(p_i^{(1)}(j)\right)_{i,j\in S} = \begin{pmatrix} 0.80 & 0.20 & 0.00 \\ 0.15 & 0.60 & 0.25 \\ 0.10 & 0.20 & 0.70 \end{pmatrix},$$

and for Selection 2:

$$P^{(2)} = \begin{pmatrix} 0.70 & 0.25 & 0.05 \\ 0.10 & 0.70 & 0.20 \\ 0.05 & 0.10 & 0.85 \end{pmatrix}.$$

Interpretation.

- The sets H_{ij} encode uncertainty or variability in transition probabilities.
- Choosing different p_i corresponds to different plausible "worlds" or expert opinions.
- Each $P^{(k)}$ is a valid stochastic matrix (rows sum to 1), recovering a classical Markov chain.
- This demonstrates the *hyper* nature: one hyperchain encodes multiple classical chains.

The definition of a Markov n-SuperHyperChain is given below.

Definition 17 (Markov n-SuperHyperChain)**.** Let S be a nonempty countable (or finite) set (the *state space*) and fix an integer $n \geq 1$. A *Markov n-SuperHyperChain* is a pair

$$\mathrm{MHC}^{(n)} \;=\; \big(S,\; H^{(n)}\big),$$

where

$$H^{(n)} \;:\; S \times S \;\longrightarrow\; \mathcal{P}_n([0,1]), \qquad (i,j) \;\mapsto\; H_{ij}^{(n)},$$

is called the *n-superhypertransition mapping*, satisfying:

(i) **Nonemptiness:**

$$H_{ij}^{(n)} \;\neq\; \varnothing \quad \forall\, i,j \in S.$$

(ii) **Nested Selection and Normalization:** For each $i \in S$, there exist *nested selection functions* $\sigma_i^{(k)}$ for $k = n, n-1, \ldots, 1$ such that:

$$\begin{aligned} &\sigma_i^{(n)} : S \;\to\; \mathcal{P}_{n-1}([0,1]), && \sigma_i^{(n)}(j) \in H_{ij}^{(n)}, \\ &\sigma_i^{(k)} : S \;\to\; \mathcal{P}_{k-1}([0,1]), && \sigma_i^{(k)}(j) \in \sigma_i^{(k+1)}(j) \quad (1 < k < n), \\ &\sigma_i^{(1)} : S \;\to\; [0,1], && \sigma_i^{(1)}(j) \in \sigma_i^{(2)}(j), \end{aligned}$$

and finally the *normalization*

$$\sum_{j\in S} \sigma_i^{(1)}(j) \;=\; 1. \tag{1}$$

Remark 6. Since $H^{(n)}$ takes values in the n-th powerset $\mathcal{P}_n([0,1])$, it is by definition an (m,n)-superhyperoperation (with $m = 2$). Hence $\mathrm{MHC}^{(n)}$ is an n-Superhyperstructure on S.

Theorem 7 (Recovery of Lower Levels)**.** *1. When $n = 1$, $\mathrm{MHC}^{(1)}$ coincides with the Markov HyperChain $\mathrm{MHC} = (S, H)$ via $H_{ij}^{(1)} = H_{ij} \subseteq [0,1]$.*

2. When $n = 0$, interpreting $\mathcal{P}_0([0,1]) = [0,1]$, $\mathrm{MHC}^{(0)}$ is exactly a classical time-homogeneous Markov chain with numerical transition matrix $P(i,j) = H_{ij}^{(0)}$.

Proof: *Case* $n = 1$. By definition $\mathcal{P}_1([0,1]) = P([0,1])$. The single nested selector $\sigma_i^{(1)} : S \to [0,1]$ with $\sigma_i^{(1)}(j) \in H_{ij}^{(1)}$ and $\sum_j \sigma_i^{(1)}(j) = 1$ is exactly the condition for a Markov HyperChain.

Case $n = 0$. Set $\mathcal{P}_0([0,1]) = [0,1]$. Then $H_{ij}^{(0)} \in [0,1]$, and the single selector $\sigma_i^{(1)} = \mathrm{id}$ yields $\sum_j H_{ij}^{(0)} = 1$, recovering the usual transition probabilities $P(i,j)$.

Theorem 8 (*n*-SuperHyperstructure Property). $\mathrm{MHC}^{(n)} = (S, H^{(n)})$ *is an* (m,n)*-Superhyperstructure with* $m = 2$.

Proof: By construction $H^{(n)} : S \times S \to \mathcal{P}_n([0,1])$. Thus $H^{(n)}$ is a binary operation whose codomain is the n-th powerset of $[0,1]$, exactly the definition of a binary (m,n)-superhyperoperation. The nested selectors guarantee that one can extract a genuine probability distribution at the bottom level, satisfying all axioms of a classical Markov chain.

Example 8 (Machine Reliability as a 2-SuperHyperChain). Machine reliability measures the probability that a machine performs its intended function without failure over a specific time period (cf.[20, 21, 22]). Consider a machine whose daily condition is modeled by the state space

$$S = \{W,\ F\}, \quad W = \text{Working},\ F = \text{Failed}.$$

We define a 2-superhypertransition

$$H^{(2)} : S \times S \longrightarrow \mathcal{P}_2([0,1])$$

by the following table of subsets of subsets of $[0,1]$:

	$j = W$	$j = F$
$i = W$	$\big\{\{0.85, 0.90\},\ \{0.80, 0.95\}\big\}$	$\big\{\{0.10, 0.15\},\ \{0.05, 0.20\}\big\}$
$i = F$	$\big\{\{0.25, 0.35\},\ \{0.30, 0.40\}\big\}$	$\big\{\{0.65, 0.80\},\ \{0.60, 0.70\}\big\}$

Nested Selection $(k = 2, 1)$**.** We choose for each source state $i \in \{W, F\}$ nested selectors $\sigma_i^{(2)} \in \mathcal{P}_1([0,1])$ and $\sigma_i^{(1)} \in [0,1]$ as follows:

	$\sigma_i^{(2)}(W)$	$\sigma_i^{(2)}(F)$
$i = W$	$\{0.85, 0.90\} \in H_{W,W}^{(2)}$,	$\{0.10, 0.15\} \in H_{W,F}^{(2)}$,
$i = F$	$\{0.25, 0.35\} \in H_{F,W}^{(2)}$,	$\{0.65, 0.80\} \in H_{F,F}^{(2)}$.

Then pick

$$\sigma_W^{(1)}(W) = 0.90 \in \{0.85, 0.90\}, \quad \sigma_W^{(1)}(F) = 0.10 \in \{0.10, 0.15\},$$

and

$$\sigma_F^{(1)}(W) = 0.35 \in \{0.25, 0.35\}, \quad \sigma_F^{(1)}(F) = 0.65 \in \{0.65, 0.80\}.$$

One checks $\sigma_W^{(1)}(W) + \sigma_W^{(1)}(F) = 0.90 + 0.10 = 1$ and $\sigma_F^{(1)}(W) + \sigma_F^{(1)}(F) = 0.35 + 0.65 = 1$, so nested normalization (1) holds.

Induced Classical Chain. The bottom-level selector $\sigma^{(1)}$ yields the classical transition matrix

$$P(i,j) = \sigma_i^{(1)}(j), \quad i, j \in \{W, F\},$$

namely

$$P = \begin{pmatrix} 0.90 & 0.10 \\ 0.35 & 0.65 \end{pmatrix}.$$

Thus the 2-SuperHyperChain $\mathrm{MHC}^{(2)}$ encodes two levels of uncertainty—choices of probability-sets—and recovers a genuine Markov chain upon nested selection.

Example 9 (Weather Forecasting Ensemble as a 3-SuperHyperChain). We model daily weather with state space

$$S = \{S, R\}, \quad S = \text{Sunny}, \ R = \text{Rainy}.$$

Fix three levels of uncertainty: *Model Family* (level 3) → *Parameter Set* (level 2) → *Probability Choice* (level 1).

Level 3: Superhypertransition sets Define $H_{ij}^{(3)} \subseteq \mathcal{P}_2([0,1])$ by two "model families" for each $i \to j$:

	$j = S$	$j = R$
$i = S$	$\left\{ F_1^{(3)}, F_2^{(3)} \right\}$	$\left\{ G_1^{(3)}, G_2^{(3)} \right\}$
$i = R$	$\left\{ H_1^{(3)}, H_2^{(3)} \right\}$	$\left\{ K_1^{(3)}, K_2^{(3)} \right\}$

where, for instance,

$$F_1^{(3)} = \big\{\{0.7, 0.8\}, \{0.75, 0.85\}\big\}, \quad F_2^{(3)} = \big\{\{0.65, 0.9\}, \{0.78, 0.82\}\big\},$$

and similarly $G_k^{(3)}, H_k^{(3)}, K_k^{(3)}$ each contain two subsets of $[0,1]$.

Level 2: Nested parameter-set selection Choose one parameter-set from each model family:
for $i = S$:

$$\sigma_S^{(3)}(S) = F_1^{(3)}, \quad \sigma_S^{(3)}(R) = G_2^{(3)};$$

then pick

$$\sigma_S^{(2)}(S) = \{0.7, 0.8\} \ \in F_1^{(3)}, \quad \sigma_S^{(2)}(R) = \{0.2, 0.25\} \ \in G_2^{(3)}.$$

for $i = R$:

$$\sigma_R^{(3)}(S) = H_2^{(3)}, \quad \sigma_R^{(3)}(R) = K_1^{(3)};$$

then pick

$$\sigma_R^{(2)}(S) = \{0.4, 0.5\} \ \in H_2^{(3)}, \quad \sigma_R^{(2)}(R) = \{0.3, 0.7\} \ \in K_1^{(3)}.$$

Level 1: Final probability choice and normalization Select a single probability from each parameter set so that they sum to 1:

$$\sigma_S^{(1)}(S) = 0.8 \in \{0.7, 0.8\}, \quad \sigma_S^{(1)}(R) = 0.2 \in \{0.2, 0.25\}, \quad 0.8 + 0.2 = 1.$$

$$\sigma_R^{(1)}(S) = 0.5 \in \{0.4, 0.5\}, \quad \sigma_R^{(1)}(R) = 0.5 \in \{0.3, 0.7\}, \quad 0.5 + 0.5 = 1.$$

Induced classical transition matrix From $\sigma^{(1)}$ we obtain

$$P(i,j) \ = \ \sigma_i^{(1)}(j), \quad i, j \in \{S, R\},$$

namely

$$P = \begin{pmatrix} 0.80 & 0.20 \\ 0.50 & 0.50 \end{pmatrix}.$$

Each nested selection $\sigma^{(3)}, \sigma^{(2)}, \sigma^{(1)}$ demonstrates how a 3-SuperHyperChain encodes three hierarchical layers of uncertainty—model families, parameter sets, and final probability choices—and recovers a classical Markov chain upon full selection.

For reference, an overview of Markov chains and their hyper/superhyper extensions is provided in Table 3.

Notion	Carrier	Transition object	Normalization / extraction condition
Markov Chain	state space S	$P : S \times S \to [0,1]$ (transition matrix)	$\sum_{j\in S} P(i,j) = 1$ for each $i \in S$
Markov HyperChain	state space S	$H : S \times S \to \mathcal{P}([0,1])$, with $H_{ij} \neq \varnothing$	for each $i \in S$ there exists a selector $p_i : S \to [0,1]$ such that $p_i(j) \in H_{ij}$ for all j, and $\sum_{j\in S} p_i(j) = 1$
Markov n-SuperHyperChain	state space S	$H^{(n)} : S \times S \to \mathcal{P}_n([0,1])$, with $H^{(n)}_{ij} \neq \varnothing$	for each $i \in S$ there exist nested selectors $\sigma_i^{(k)}$ $(k = n, n-1, \ldots, 1)$ such that $\sigma_i^{(n)}(j) \in H^{(n)}_{ij}$, $\sigma_i^{(k)}(j) \in \sigma_i^{(k+1)}(j)$, and $\sum_{j\in S} \sigma_i^{(1)}(j) = 1$

TABLE 3. A compact overview of Markov chains and their hyper/superhyper extensions. Here $\mathcal{P}_n([0,1])$ denotes the n-fold iterated powerset of $[0,1]$. Interpreting $\mathcal{P}_0([0,1]) = [0,1]$ yields the classical level, and $n = 1$ yields the hyper level.

2.3|HyperIntervals and SuperHyperIntervals

An interval is a nonempty subset of a totally ordered set where, for any two elements, all elements between them are also included—representing continuous ranges (cf.[23]). In this subsection, we extend this concept to HyperIntervals and SuperHyperIntervals and investigate their properties.

Definition 18 (Totally Ordered Set)**.** [24] A *totally ordered set* is a pair $(X, \leq)$ where X is a set and $\leq$ is a binary relation on X satisfying, for all $x, y, z \in X$:

(i) **Reflexivity:** $x \leq x$.

(ii) **Antisymmetry:** If $x \leq y$ and $y \leq x$, then $x = y$.

(iii) **Transitivity:** If $x \leq y$ and $y \leq z$, then $x \leq z$.

(iv) **Totality:** Either $x \leq y$ or $y \leq x$.

Definition 19 (Convex Subset)**.** Let $(X, \leq)$ be a totally ordered set. A subset $I \subseteq X$ is *convex* if for every $x, y \in I$ with $x \leq y$, any $z \in X$ satisfying $x \leq z \leq y$ also lies in I:

$$x,\, y \in I,\ x \leq z \leq y \implies z \in I.$$

Definition 20 (Interval)**.** An *interval* in a totally ordered set $(X, \leq)$ is any nonempty convex subset $I \subseteq X$. Equivalently, I is an interval if there exist (possibly equal or infinite) endpoints $a, b \in X \cup \{-\infty, +\infty\}$ with $a \leq b$ such that

$$I \;=\; \{\, x \in X \mid a \Diamond x \Diamond' b\},$$

where each of $\Diamond, \Diamond'$ is chosen from $\{<, \leq\}$. In particular, in the real line $(\mathbb{R}, \leq)$ one has:

$$[a,b] = \{x \in \mathbb{R} \mid a \leq x \leq b\},$$
$$(a,b) = \{x \in \mathbb{R} \mid a < x < b\},$$
$$[a,b) = \{x \in \mathbb{R} \mid a \leq x < b\},$$
$$(a,b] = \{x \in \mathbb{R} \mid a < x \leq b\},$$

with the conventions $(-\infty, b] = \{x \mid x \leq b\}$, $(a, \infty) = \{x \mid x > a\}$, etc.

Remark 7.

- A *degenerate interval* has $a = b$, e.g. $[a, a] = \{a\}$ (cf.[25]).
- The whole set X is an interval with $(a, b) = (-\infty, +\infty)$.
- Intervals are exactly the connected subsets of $\mathbb{R}$ under the usual topology.

Example 10. (a) $[0,1] \subset \mathbb{R}$ is a closed bounded interval.

(b) $(-\infty, 0) \subset \mathbb{R}$ is an unbounded open interval.

(c) $\{5\} = [5,5]$ is a degenerate (singleton) interval.

(d) In $\mathbb{Z}$ with the usual order, $\{2,3,4\} = [2,4] \cap \mathbb{Z}$ is an interval.

The definition of a Hyperinterval is given below.

Definition 21 (Hyperinterval Hyperoperation). Define a binary *hyperoperation*

$$\boxtimes : \mathcal{I}(X) \times \mathcal{I}(X) \longrightarrow \mathcal{P}\big(\mathcal{I}(X)\big)$$

by

$$[a_1, b_1] \boxtimes [a_2, b_2] = \big\{ \big[\min(x,y), \max(x,y)\big] \;\big|\; x \in [a_1, b_1],\ y \in [a_2, b_2] \big\}.$$

The pair $\big(\mathcal{I}(X), \boxtimes\big)$ is called the *hyperinterval hyperstructure* on X.

Remark 8. For any $x \in [a_1, b_1]$ and $y \in [a_2, b_2]$ we have $\min(x,y) \le \max(x,y)$, so each $[\min(x,y), \max(x,y)]$ is indeed a closed interval in X. Hence $\boxtimes$ maps into $\mathcal{P}(\mathcal{I}(X))$ and is a legitimate hyperoperation.

Theorem 9 (Recovery of Classical Interval Span). *For any two intervals $I = [a_1, b_1]$ and $J = [a_2, b_2]$, the smallest closed interval containing $I \cup J$,*

$$\mathrm{hull}(I, J) = \big[\min(a_1, a_2),\ \max(b_1, b_2)\big],$$

is recovered by choosing $x = \min(a_1, a_2) \in I \cup J$ and $y = \max(b_1, b_2) \in I \cup J$, so that

$$\big[\min(x,y),\ \max(x,y)\big] = \big[\min(a_1, a_2),\ \max(b_1, b_2)\big] \in I \boxtimes J.$$

Thus $\boxtimes$ genuinely generalizes the classical "interval span" operation.

Proof: Since $\min(a_1, a_2) \le \max(b_1, b_2)$, the chosen pair (x, y) satisfies $x \in [a_1, b_1] \cup [a_2, b_2]$ and $y \in [a_1, b_1] \cup [a_2, b_2]$. By the definition of $\boxtimes$,

$$[\min(x,y), \max(x,y)] = \big[\min(a_1, a_2),\ \max(b_1, b_2)\big]$$

belongs to $I \boxtimes J$. This interval is exactly the smallest closed interval containing both I and J.

Theorem 10 (Hyperstructure Property). $\big(\mathcal{I}(X), \boxtimes\big)$ *is a* hyperstructure *because its binary operation $\boxtimes$ takes each pair of intervals to a subset of $\mathcal{I}(X)$:*

$$\boxtimes : \mathcal{I}(X) \times \mathcal{I}(X) \longrightarrow \mathcal{P}\big(\mathcal{I}(X)\big).$$

Proof: By construction, for any $I, J \in \mathcal{I}(X)$ the set $I \boxtimes J$ is a (possibly empty) subset of $\mathcal{I}(X)$. No further closure or algebraic axioms are required for a hyperstructure beyond this set-valued operation.

Example 11 (Hyperinterval for Scheduling Windows). Scheduling windows are specific time intervals during which tasks or operations are allowed, planned, or expected to be executed. Let $X = \mathbb{R}$ represent time in hours and consider two scheduling windows:

$$I = [10,\ 15], \qquad J = [12,\ 20].$$

Here I might be a maintenance window from 10:00 to 15:00, and J a delivery window from 12:00 to 20:00.

Hyperinterval Computation By definition,

$$I \boxtimes J = \big\{ \big[\min(x,y),\, \max(x,y)\big] \mid x \in [10,15],\ y \in [12,20] \big\}.$$

Concretely:

- If $x = 10$ and $y = 12$, then [min, max] $= [10, 12]$.
- If $x = 10$ and $y = 20$, then [min, max] $= [10, 20]$.
- If $x = 15$ and $y = 12$, then [min, max] $= [12, 15]$.
- If $x = 15$ and $y = 20$, then [min, max] $= [15, 20]$.
- If $x = 13.5$ and $y = 14$, then [min, max] $= [13.5, 14]$.

Description of the Result One checks that

$$I \boxtimes J = \big\{ [u,v] \ \big|\ u \in [10,20],\ v \in [12,20],\ u \le v \big\}.$$

Thus:

- The *earliest possible start* is $\min(10, 12) = 10$.
- The *latest possible end* is $\max(15, 20) = 20$.
- Every interval $[u, v]$ with $10 \le u \le v \le 20$ and $v \ge 12$ appears.

Interpretation In scheduling terms, the hyperinterval captures all possible combined windows: you might start as early as 10:00 (if maintenance begins before delivery) or as late as 12:00 (if delivery precedes maintenance), and finish as early as 12:00 or as late as 20:00. Every such candidate window $[u, v]$ is a valid member of the hyperinterval $I \boxtimes J$.

The definition of a n-SuperHyperinterval is given below.

Definition 22 (*n*-th Powerset of Intervals). Let $(X, \le)$ be a totally ordered set and let

$$\mathcal{I}(X) = \{\, [a,b] \mid a, b \in X \cup \{-\infty, +\infty\},\ a \le b \}$$

be the set of all closed intervals in X. Define recursively

$$\mathcal{I}_0(X) = \mathcal{I}(X), \quad \mathcal{I}_{k+1}(X) = \mathcal{P}\big(\mathcal{I}_k(X)\big), \quad k \ge 0.$$

Definition 23 (n-SuperHyperinterval Hyperoperation). For each integer $n \ge 1$, define a binary hyperoperation

$$\boxtimes^{(n)} : \mathcal{I}_n(X) \times \mathcal{I}_n(X) \ \longrightarrow\ \mathcal{P}\big(\mathcal{I}_n(X)\big)$$

by recursion on n. Given $A, B \in \mathcal{I}_n(X)$,

$$A \boxtimes^{(n)} B = \big\{ U \boxtimes^{(n-1)} V \mid U \in A,\ V \in B \big\},$$

where the *base case* $n = 0$ is the classical interval span:

$$I \boxtimes^{(0)} J = [\min(a_1, a_2),\, \max(b_1, b_2)], \quad I = [a_1, b_1],\ J = [a_2, b_2] \in \mathcal{I}(X).$$

Definition 24 (*n*-SuperHyperinterval Hyperstructure). The pair

$$\big(\mathcal{I}_n(X),\, \boxtimes^{(n)}\big)$$

is called the *n-SuperHyperinterval Hyperstructure* on X.

Remark 9. Since $\boxtimes^{(n)}$ takes its values in $\mathcal{P}(\mathcal{I}_n(X))$, it is a binary $(2, n)$-superhyperoperation. Hence $\big(\mathcal{I}_n(X), \boxtimes^{(n)}\big)$ is an n-Superhyperstructure.

Theorem 11 (Recovery of Lower Levels). *(i) For $n = 0$, $(\mathcal{I}_0(X), \boxtimes^{(0)})$ is the classical interval span.*

(ii) For $n = 1$, $(\mathcal{I}_1(X), \boxtimes^{(1)})$ coincides with the hyperinterval hyperstructure $(\mathcal{I}(X), \boxtimes)$.

Proof: *(i)* By definition $\mathcal{I}_0(X) = \mathcal{I}(X)$ and $\boxtimes^{(0)}$ is exactly $I \boxtimes^{(0)} J = [\min(a_1, a_2), \max(b_1, b_2)]$.

(ii) Since $\mathcal{I}_1(X) = \mathcal{P}(\mathcal{I}(X))$, the recursion gives

$$A \boxtimes^{(1)} B = \{\, I \boxtimes^{(0)} J \mid I \in A,\ J \in B\},$$

which is precisely the hyperinterval operation $\boxtimes$.

Theorem 12 (n-Superhyperstructure Property). $\left(\mathcal{I}_n(X), \boxtimes^{(n)}\right)$ *is an n-Superhyperstructure, i.e. its operation has codomain $\mathcal{P}(\mathcal{I}_n(X))$.*

Proof: By construction $\mathcal{I}_n(X) = \mathcal{P}(\mathcal{I}_{n-1}(X))$, so for any $A, B \in \mathcal{I}_n(X)$, the set $A \boxtimes^{(n)} B$ belongs to $\mathcal{P}(\mathcal{I}_n(X))$, verifying the superhyperstructure condition.

Example 12 (2-SuperHyperinterval: Scheduling and Delivery Windows). Let $X = \mathbb{R}$ represent time in hours. Consider three sets of intervals:

$$S_1 = \{[9, 11],\, [14, 16]\}, \quad S_2 = \{[10, 12],\, [15, 18]\}, \quad S_3 = \{[11, 13]\}.$$

Each $S_k \subseteq \mathcal{I}(X)$, so

$$A = \{\, S_1, S_2\} \ \in\ \mathcal{I}_2(X), \qquad B = \{\, S_3\} \ \in\ \mathcal{I}_2(X).$$

By definition,

$$A \boxtimes^{(2)} B = \{\, U \boxtimes^{(1)} V \mid U \in A,\ V \in B\}.$$

We compute each:

$$\begin{aligned} S_1 \boxtimes^{(1)} S_3 &= \{\, I \boxtimes^{(0)} J \mid I \in S_1,\ J \in S_3\} \\ &= \{[9, 11] \boxtimes^{(0)} [11, 13],\ [14, 16] \boxtimes^{(0)} [11, 13]\} \\ &= \{[\,\min(9, 11),\ \max(11, 13)\,],\ [\,\min(14, 11),\ \max(16, 13)\,]\} \\ &= \{[9, 13],\, [11, 16]\}. \end{aligned}$$

$$\begin{aligned} S_2 \boxtimes^{(1)} S_3 &= \{\, I \boxtimes^{(0)} J \mid I \in S_2,\ J \in S_3\} \\ &= \{[10, 12] \boxtimes^{(0)} [11, 13],\ [15, 18] \boxtimes^{(0)} [11, 13]\} \\ &= \{[10, 13],\, [11, 18]\}. \end{aligned}$$

Therefore the full 2-superhyperinterval is

$$A \boxtimes^{(2)} B = \big\{\{[9, 13], [11, 16]\},\ \{[10, 13], [11, 18]\}\big\}\ \subseteq\ \mathcal{I}_2(X).$$

Interpretation:

- S_1 might represent two possible maintenance windows (9–11 h, 14–16 h).
- S_2 two delivery windows (10–12 h, 15–18 h), and S_3 a fixed inspection window (11–13 h).
- The first-level operation $\boxtimes^{(1)}$ combines each maintenance or delivery interval with the inspection window, yielding sets of possible combined windows.
- The second-level operation $\boxtimes^{(2)}$ collects these first-level results into a set of two candidate sets, reflecting two different scheduling scenarios.

Example 13 (A Real-World 3-SuperHyperinterval: Multi-Tier Scheduling). Multi-Tier Scheduling coordinates tasks across hierarchical levels—individual, team, and organizational—ensuring alignment, resource availability, and timing at each layer (cf.[26, 27, 27]). Let $X = \mathbb{R}$ be time in hours. We illustrate three hierarchical scheduling levels:

Level 0 (Basic Intervals):

$$S_1 = [9, 11], \quad S_2 = [14, 16], \quad S_3 = [10, 12], \quad S_4 = [15, 18], \quad S_5 = [11, 13].$$

These lie in $\mathcal{I}_0(X) = \mathcal{I}(X)$.

Level 1 (Team-Level Sets): Form two teams' windows:

$$U = \{S_1, S_2\}, \quad V = \{S_3, S_4\}, \quad W = \{S_5\} \ \in \ \mathcal{I}_1(X) = \mathcal{P}(\mathcal{I}(X)).$$

Level 2 (Department-Level Collections): Group these team-sets into departments:

$$D_1 = \{U, V\}, \quad D_2 = \{W\} \ \in \ \mathcal{I}_2(X) = \mathcal{P}(\mathcal{I}_1(X)).$$

Level 3 (Division-Level Supercollections): Finally, form two divisions' supercollections:

$$A = \{D_1\}, \quad B = \{D_2\} \ \in \ \mathcal{I}_3(X) = \mathcal{P}(\mathcal{I}_2(X)).$$

Third-Level Hyperoperation $\boxtimes^{(3)}$:

$$A \boxtimes^{(3)} B = \{\, D_1 \boxtimes^{(2)} D_2\}.$$

Compute the inner 2-superhyperinterval:

$$D_1 \boxtimes^{(2)} D_2 = \big\{\, U \boxtimes^{(1)} W,\ V \boxtimes^{(1)} W\big\},$$

where each $\boxtimes^{(1)}$ is the hyperinterval on sets of intervals:

$$\begin{aligned} U \boxtimes^{(1)} W &= \{\, I \boxtimes^{(0)} J \mid I \in U,\ J \in W\} \\ &= \{\, S_1 \boxtimes^{(0)} S_5,\ S_2 \boxtimes^{(0)} S_5\} \\ &= \{[9, 13],\ [11, 16]\}, \\ V \boxtimes^{(1)} W &= \{\, S_3 \boxtimes^{(0)} S_5,\ S_4 \boxtimes^{(0)} S_5\} \\ &= \{[10, 13],\ [11, 18]\}. \end{aligned}$$

Thus

$$D_1 \boxtimes^{(2)} D_2 = \big\{\{[9, 13], [11, 16]\},\ \{[10, 13], [11, 18]\}\big\},$$

and finally

$$A \boxtimes^{(3)} B = \big\{\, D_1 \boxtimes^{(2)} D_2\big\} = \Big\{\{\{[9, 13], [11, 16]\}, \{[10, 13], [11, 18]\}\}\Big\}.$$

Interpretation:

- Level 1 intervals S_i are individual team work or delivery windows.
- Level 1 sets U, V, W group those into team schedules.
- Level 2 collections D_1, D_2 are department-wide scheduling options.
- Level 3 supercollections A, B represent division-level bundles.
- The 3-superhyperinterval $A \boxtimes^{(3)} B$ encodes all possible combined division schedules across three hierarchical tiers.

For reference, Table 4 presents an overview of intervals and their hyper/superhyper extensions.

Concept	Object and notation	Extension mechanism (informal)
Interval	Nonempty convex $I \subseteq (X, \leq)$ (e.g., in $\mathbb{R}$: $I = [a, b]$).	Order-convexity: $x, y \in I,\ x \leq z \leq y \Rightarrow z \in I$.
HyperInterval	Universe $\mathcal{I}(X)$. Hyperoperation $I \boxtimes J := \{[\min(x, y), \max(x, y)] \mid x \in I,\ y \in J\} \subseteq \mathcal{I}(X)$.	Replace the single span by a set of spans obtained from internal choices $x \in I$, $y \in J$.
n-SuperHyperInterval	Iteration: $\mathcal{I}_0(X) = \mathcal{I}(X)$, $\mathcal{I}_{k+1}(X) = \mathcal{P}(\mathcal{I}_k(X))$. For $A, B \in \mathcal{I}_n(X)$, $A \boxtimes^{(n)} B := \{ U \boxtimes^{(n-1)} V \mid U \in A,\ V \in B \}$. (Base $n = 0$: $I \boxtimes^{(0)} J = [\min(a_1, a_2), \max(b_1, b_2)]$.)	Lift the operation level-by-level: objects are sets of lower-level objects; combine by taking all memberwise combinations recursively.

Table 4. Intervals and their hyper/superhyper extensions.

2.4|Abstract Propositional HyperLogic and Superhyperlogic

Propositional logic analyzes truth values of propositions using logical connectives such as AND, OR, and NOT, without considering internal structure (cf.[28, 29]). Abstract propositional logic studies logical consequence relations over arbitrary sets of formulas, independently of any specific syntax or semantics. In this paper, we extend this framework to *Abstract Propositional HyperLogic* and *Superhyperlogic* by employing HyperStructures and SuperHyperStructures.

Definition 25 (Language of Propositional Logic). Let *Var* be a nonempty set of *propositional variables*. The *language* $\mathcal{L}_{\text{prop}}$ is the smallest set of strings (called *formulas*) such that:

$$\varphi \ ::= \ p \quad (p \in \mathit{Var}) \ \Big| \ (\neg\varphi) \ \Big| \ (\varphi \wedge \psi) \ \Big| \ (\varphi \vee \psi) \ \Big| \ (\varphi \to \psi),$$

where $\varphi, \psi \in \mathcal{L}_{\text{prop}}$. We write $\mathcal{L}_{\text{prop}}(\mathit{Var})$ when we wish to emphasize the dependence on *Var*.

Definition 26 (Abstract Logic). An *abstract propositional logic* is a pair

$$L \ = \ \big(\mathcal{L}_{\text{prop}}, \vdash_L\big),$$

where

- $\mathcal{L}_{\text{prop}}$ is the set of well-formed formulas (as above).
- $\vdash_L \subseteq \mathcal{P}(\mathcal{L}_{\text{prop}}) \times \mathcal{L}_{\text{prop}}$ is a *consequence relation* satisfying, for all $\Gamma, \Delta \subseteq \mathcal{L}_{\text{prop}}$ and $\varphi, \psi \in \mathcal{L}_{\text{prop}}$:
  (1) **Reflexivity:** If $\varphi \in \Gamma$, then $\Gamma \vdash_L \varphi$.
  (2) **Monotonicity:** If $\Gamma \vdash_L \varphi$ and $\Gamma \subseteq \Delta$, then $\Delta \vdash_L \varphi$.
  (3) **Cut (Transitivity):** If $\Gamma \vdash_L \psi$ for every $\psi \in \Delta$ and $\Delta \vdash_L \varphi$, then $\Gamma \vdash_L \varphi$.
  (4) **Substitution-Invariance:** If $\Gamma \vdash_L \varphi$ then $\sigma(\Gamma) \vdash_L \sigma(\varphi)$ for every uniform substitution σ on propositional variables.

Definition 27 (Semantic Consequence). Let $L = (\mathcal{L}_{\text{prop}}, \vdash_L)$ be an abstract logic. A *valuation* is a function $v : \mathit{Var} \to \{0, 1\}$, extended to all of $\mathcal{L}_{\text{prop}}$ by:

$$v(\neg\varphi) = 1 - v(\varphi), \quad v(\varphi \wedge \psi) = \min\{v(\varphi), v(\psi)\},$$

and similarly for $\vee$ and $\to$. We write

$$\Gamma \models \varphi \quad \Longleftrightarrow \quad \text{for every valuation } v,\ \big(v(\psi) = 1\ \forall \psi \in \Gamma\big) \text{ implies } v(\varphi) = 1.$$

Definition 28 (Soundness and Completeness)**.** An abstract logic $L = (\mathcal{L}_{\text{prop}}, \vdash_L)$ is called

- *sound* if $\Gamma \vdash_L \varphi$ implies $\Gamma \models \varphi$.
- *complete* if $\Gamma \models \varphi$ implies $\Gamma \vdash_L \varphi$.

The definition of an Abstract Hyperlogic is given below.

Definition 29 (Abstract Hyperlogic)**.** Let Var be a nonempty set of propositional variables, and let $\mathcal{L} = \mathcal{L}_{\text{prop}}(Var)$ be the set of all well-formed propositional formulas over Var. Set

$$H \;=\; \mathcal{P}(\mathcal{L}),$$

the collection of all sets of formulas. An *abstract hyperlogic* is a pair

$$\big(H,\ \dashv_H\big),$$

where

$$\dashv_H :\ H \;\longrightarrow\; \mathcal{P}(H)$$

is a *hyperconsequence operator* (a unary hyperoperation) satisfying, for all $\Gamma, \Delta, \Theta \in H$:

(a) **Hyperreflexivity:** $\Gamma \;\in \dashv_H (\Gamma)$.

(b) **Hypermonotonicity:** If $\Delta \in \dashv_H (\Gamma)$ and $\Delta \subseteq \Theta$, then $\Theta \in \dashv_H (\Gamma)$.

(c) **Hypercut:** If $\Delta \in \dashv_H (\Gamma)$ and $\Theta \in \dashv_H (\Delta)$, then $\Theta \in \dashv_H (\Gamma)$.

Remark 10 (Hyperstructure Property)**.** Since $\dashv_H$: $H \to \mathcal{P}(H)$, the pair $(H, \dashv_H)$ is by definition a *hyperstructure* with a unary hyperoperation.

Theorem 13 (Recovery of Abstract Logic)**.** *If there exists a classical consequence relation* $\vdash \subseteq H \times \mathcal{L}$ *whose associated closure operator*

$$\mathrm{Cl} : H \longrightarrow H, \quad \mathrm{Cl}(\Gamma) = \{\varphi \in \mathcal{L} \mid \Gamma \vdash \varphi\},$$

then setting

$$\dashv_H (\Gamma) \;=\; \big\{\mathrm{Cl}(\Gamma)\big\} \quad (\textit{a singleton set})$$

yields an abstract hyperlogic $(H, \dashv_H)$ *which exactly generalizes the original logic.*

Proof: Define $\dashv_H (\Gamma) = \{\mathrm{Cl}(\Gamma)\}$. Then for each Γ:

- *Hyperreflexivity:* $\Gamma \subseteq \mathrm{Cl}(\Gamma)$, so $\Gamma \in \dashv_H (\Gamma)$.
- *Hypermonotonicity:* If $\mathrm{Cl}(\Gamma) \subseteq \Theta$, then by monotonicity of Cl, $\Theta = \mathrm{Cl}(\Gamma)$, so $\Theta \in \dashv_H (\Gamma)$.
- *Hypercut:* Since $\dashv_H (\Gamma)$ and $\dashv_H (\Delta)$ are singletons, the transitivity of Cl ensures closure under two-step application.

Thus $\dashv_H$ satisfies (a)–(c) and recovers the original $\vdash$.

Example 14 (Concrete Nontrivial Hyperlogic)**.** Let $Var = \{p, q\}$ and consider the finite fragment of the propositional language

$$\mathcal{L}' = \big\{\, p,\ q,\ p \to q,\ q \to p \big\}.$$

Fix the premise set

$$\Gamma \;=\; \{\, p \to q,\ p\}.$$

We define a hyperconsequence operator $\dashv_H$: $\mathcal{P}(\mathcal{L}') \to \mathcal{P}^2(\mathcal{L}')$ by

$$\dashv_H (\Gamma) = \big\{\Gamma,\ \underbrace{\Gamma \cup \{q\}}_{=C},\ \underbrace{\mathcal{L}'}_{=T}\big\}.$$

Contents of each hyperconsequence set:

$$\Gamma = \{p \to q,\ p\}, \quad C = \{p \to q,\ p,\ q\}, \quad T = \{p,\ q,\ p \to q,\ q \to p\}.$$

Verification of hyperlogic axioms:

- *Hyperreflexivity:* $\Gamma \in \dashv_H (\Gamma)$ by construction.
- *Hypermonotonicity:* Any superset of Γ in $\mathcal{L}'$ is either C or T, and both lie in $\dashv_H (\Gamma)$. Any superset of C is T, also in $\dashv_H (\Gamma)$.
- *Hypercut:* We need that if $\Delta \in \dashv_H (\Gamma)$ and $\Theta \in \dashv_H (\Delta)$, then $\Theta \in \dashv_H (\Gamma)$.

– If $\Delta = \Gamma$, then $\dashv_H (\Delta) = \{\Gamma, C, T\}$, so any $\Theta \in \dashv_H (\Gamma)$.

– If $\Delta = C$, then $\dashv_H (C) = \{C, T\}$ (by the same rule), and both lie in $\dashv_H (\Gamma)$.

– If $\Delta = T$, then $\dashv_H (T) = \{T\}$, and $T \in \dashv_H (\Gamma)$.

Interpretation:

- Γ itself as a minimal consequence set (no inference).
- C is the classical propositional closure of Γ, since from $p \to q$ and p we infer q.
- T is the "explosive" or trivial logic closure, in which every formula follows.

Thus $\dashv_H (\Gamma)$ contains three nested consequence-sets, demonstrating genuine hyperbehavior beyond any single abstract logic.

The definition of an Abstract n-SuperHyperlogic is given below.

Definition 30 (Abstract n-SuperHyperlogic)**.** Fix $n \geq 0$. Define

$$\delta^{(0)} : D_0 \longrightarrow D_0, \qquad \delta^{(0)}(\Gamma) = \mathrm{Cl}(\Gamma),$$

and for $k = 0, 1, \ldots, n-1$,

$$\delta^{(k+1)} : D_0 \longrightarrow D_{k+1}, \qquad \delta^{(k+1)}(\Gamma) = \left\{\delta^{(k)}(\Gamma)\right\}.$$

Then the pair

$$\left(D_0,\ \delta^{(n)} \right)$$

is called the *abstract n-superhyperlogic*.

Remark 11 (Superhyperstructure Property). Since $\delta^{(n)}$ maps $D_0 = \mathcal{P}(\mathcal{L})$ into the n-fold powerset $D_n = \mathcal{P}^{n+1}(\mathcal{L})$, it is a unary $(1, n)$-superhyperoperation, and thus $\left(D_0, \delta^{(n)}\right)$ is a $(1, n)$-superhyperstructure.

Theorem 14 (Recovery of Logic and Hyperlogic)**.** *(i) If $n = 0$, then $\delta^{(0)} = \mathrm{Cl}$ and $\left(D_0, \delta^{(0)}\right)$ is exactly the original abstract logic.*

(ii) If $n = 1$, then $\delta^{(1)}(\Gamma) = \{\mathrm{Cl}(\Gamma)\}$, so $\left(D_0, \delta^{(1)}\right)$ coincides with the abstract hyperlogic.

Proof: • For $n = 0$, by definition $\delta^{(0)}(\Gamma) = \mathrm{Cl}(\Gamma)$.

- For $n = 1$,
$$\delta^{(1)}(\Gamma) = \{\delta^{(0)}(\Gamma)\} = \{\mathrm{Cl}(\Gamma)\},$$
recovering the singleton-valued hyperconsequence.

Higher levels follow by the inductive construction of $\delta^{(k)}$.

Theorem 15 (Nested Expansion). *For every $\Gamma \subseteq \mathcal{L}$ and every $n \geq 0$,*

$$\delta^{(n)}(\Gamma) \ = \ \underbrace{\{\{\cdots\{\mathrm{Cl}(\Gamma)\}\cdots\}\}}_{n \text{ neste}}.$$

Proof: We proceed by induction on n.

Base case ($n = 0$). By definition $\delta^{(0)}(\Gamma) = \mathrm{Cl}(\Gamma)$, which is just one application of the closure.

Inductive step. Suppose the formula holds for some $n = k$, i.e.

$$\delta^{(k)}(\Gamma) = \underbrace{\{\{\cdots\{\mathrm{Cl}(\Gamma)\}\cdots\}\}}_{k \text{ braces}}.$$

Then by definition of $\delta^{(k+1)}$,

$$\delta^{(k+1)}(\Gamma) = \{\delta^{(k)}(\Gamma)\} = \Big\{\underbrace{\{\{\cdots\{\mathrm{Cl}(\Gamma)\}\cdots\}\}}_{k \text{ braces}}\Big\},$$

which is exactly $k+1$ nested braces around $\mathrm{Cl}(\Gamma)$. This completes the induction.

Theorem 15 (Idempotence)**.** *For every $\Gamma \subseteq \mathcal{L}$ and every $n \geq 0$,*

$$\delta^{(n)}\big(\delta^{(n)}(\Gamma)\big) = \delta^{(n)}(\Gamma).$$

Proof: Again by induction on n.

Base case ($n = 0$).

$$\delta^{(0)}\big(\delta^{(0)}(\Gamma)\big) = \mathrm{Cl}\big(\mathrm{Cl}(\Gamma)\big) = \mathrm{Cl}(\Gamma) = \delta^{(0)}(\Gamma),$$

using idempotence of the Tarskian closure Cl.

Inductive step. Assume $\delta^{(k)}$ is idempotent. Then

$$\delta^{(k+1)}\big(\delta^{(k+1)}(\Gamma)\big) = \Big\{\delta^{(k)}\big(\delta^{(k+1)}(\Gamma)\big)\Big\} = \Big\{\delta^{(k)}(\Gamma)\Big\} = \delta^{(k+1)}(\Gamma),$$

where the middle equality uses the fact that $\delta^{(k+1)}(\Gamma) = \{\delta^{(k)}(\Gamma)\}$ and the inductive hypothesis $\delta^{(k)}(\delta^{(k)}(\Gamma)) = \delta^{(k)}(\Gamma)$.

Theorem 16 (Substitution Invariance)**.** *Let σ be any uniform substitution on propositional variables, extended pointwise to sets and iterated powersets. Then for every $\Gamma \subseteq \mathcal{L}$ and every $n \geq 0$,*

$$\delta^{(n)}\big(\sigma(\Gamma)\big) = \sigma\big(\delta^{(n)}(\Gamma)\big),$$

where on the right σ acts level-wise on each nested set.

Proof: We prove by induction on n.

Base case ($n = 0$).

$$\delta^{(0)}\big(\sigma(\Gamma)\big) = \mathrm{Cl}\big(\sigma(\Gamma)\big) = \sigma\big(\mathrm{Cl}(\Gamma)\big) = \sigma\big(\delta^{(0)}(\Gamma)\big),$$

using substitution-invariance of the closure operator Cl.

Inductive step. Suppose the claim holds for $n = k$. Then

$$\delta^{(k+1)}\big(\sigma(\Gamma)\big) = \Big\{\delta^{(k)}\big(\sigma(\Gamma)\big)\Big\} = \Big\{\sigma\big(\delta^{(k)}(\Gamma)\big)\Big\} = \sigma\Big(\{\delta^{(k)}(\Gamma)\}\Big) = \sigma\big(\delta^{(k+1)}(\Gamma)\big),$$

where we used the inductive hypothesis in the second equality and the natural action of σ on singleton-wrapped sets in the third.

Example 15 (Nested Consequences: Multi-Step Inference). Let $Var = \{p, q, r\}$ and let $\mathcal{L}$ be the set of all propositional formulas built with $\wedge, \vee, \rightarrow, \neg$. Let Cl be the classical propositional closure operator (closing under modus ponens, conjunction introduction, etc.). Consider the premise set

$$\Gamma = \{\, p \rightarrow q,\ q \rightarrow r,\ p\}.$$

- $\delta^{(0)}(\Gamma) = \mathrm{Cl}(\Gamma)$ is the set of all formulas derivable from Γ. In particular:

  $$\mathrm{Cl}(\Gamma) = \{\, p \rightarrow q,\ q \rightarrow r,\ p,\ q,\ r\}.$$

  Indeed, from $p \rightarrow q$ and p we infer q, and from $q \rightarrow r$ and q we infer r.

- $\delta^{(1)}(\Gamma)$ lifts this to a singleton set:

  $$\delta^{(1)}(\Gamma) = \big\{\ \mathrm{Cl}(\Gamma)\big\} = \big\{\{p \rightarrow q,\ q \rightarrow r,\ p,\ q,\ r\}\big\}.$$

- $\delta^{(2)}(\Gamma)$ lifts one more level:

  $$\delta^{(2)}(\Gamma) = \big\{\ \delta^{(1)}(\Gamma)\big\} = \Big\{\{\{p \rightarrow q,\ q \rightarrow r,\ p,\ q,\ r\}\}\Big\}.$$

Thus each application of δ moves the consequence set up one level in the powerset hierarchy, giving a fully detailed illustration of the 2-superhyperlogic for a nontrivial inference scenario.

Example 2.65 (Medical Diagnosis Workflow as a 3-SuperHyperlogic)**.** Medical diagnosis identifies a patient's disease or condition by analyzing symptoms, history, tests, and observations using clinical and scientific methods. Let

$$Var = \{Fever,\ Cough\}, \quad \mathcal{L} = \mathcal{L}_{\text{prop}}(Var),$$

and introduce two diagnostic conclusions:

$$D_1 \equiv \text{“Diagnosis: Flu”}, \quad D_2 \equiv \text{“Diagnosis: Common Cold”}.$$

Define the Tarskian closure Cl by medical inference rules so that

$$\text{Cl}\big(\{Fever, Cough\}\big) = \{D_1, D_2\}.$$

Set $\Gamma = \{Fever, Cough\}$. Then the 3-superhyperlogic layers are:

$$\delta^{(0)}(\Gamma) = \text{Cl}(\Gamma) = \{D_1, D_2\},$$

$$\delta^{(1)}(\Gamma) = \big\{\delta^{(0)}(\Gamma)\big\} = \big\{\{D_1, D_2\}\big\},$$

$$\delta^{(2)}(\Gamma) = \big\{\delta^{(1)}(\Gamma)\big\} = \big\{\{\{D_1, D_2\}\}\big\},$$

$$\delta^{(3)}(\Gamma) = \big\{\delta^{(2)}(\Gamma)\big\} = \big\{\{\{\{D_1, D_2\}\}\}\big\}.$$

Interpretation

- $\delta^{(0)}$ yields the direct diagnostic conclusions from the symptoms.
- $\delta^{(1)}$ wraps those conclusions to represent a *primary review* (e.g., by a general practitioner).
- $\delta^{(2)}$ nests one more level to model a *secondary review* (e.g., by a specialist), encapsulating the primary's output.
- $\delta^{(3)}$ adds a *tertiary expert panel* validation layer, nesting all prior reviews.

Thus the medical diagnosis process—symptoms → primary diagnosis → specialist review → panel validation—is captured as a 3-SuperHyperlogic with three tiers of nested inference.

For reference, Table 5 presents an overview of abstract propositional logic and its hyper/superhyper extensions.

2.5|Discrete-Time HyperSystem and n-SuperHyperSystem

A discrete-time system evolves in fixed time steps, updating states based on inputs or previous states at each time index (cf.[30, 31, 32]). In this subsection, we define Discrete-Time HyperSystems and n-SuperHyperSystems.

Definition 31 (Discrete-Time System)**.** A *(discrete-time) system* is a tuple

$$S = \big(E,\ I,\ X,\ T,\ t_0,\ x,\ F\big),$$

where:

- E is a nonempty set of *components*.
- $I \subseteq E \times E$ is the *influence relation*: $(j, i) \in I$ means component j influences component i.
- X is the *state space*.
- T is a nonempty totally ordered set (time index).
- $t_0 = \min T$ is the initial time.

Concept	Abstract Propositional Logic	Abstract Propositional HyperLogic	Abstract Propositional SuperHyperLogic
Carrier (universe)	$\mathcal{L} = \mathcal{L}_{\text{prop}}(Var)$ (formulas) and $H = \mathcal{P}(\mathcal{L})$ (theories/premise sets)	$H = \mathcal{P}(\mathcal{L})$	$D_0 = \mathcal{P}(\mathcal{L})$, and $D_{k+1} = \mathcal{P}(D_k) = \mathcal{P}^{k+2}(\mathcal{L})$
Core inference object	Consequence relation $\vdash_L \subseteq \mathcal{P}(\mathcal{L}) \times \mathcal{L}$	Hyperconsequence operator (unary hyperoperation) $\dashv_H$: $H \to \mathcal{P}(H)$	Iterated consequence lifting $\delta^{(n)} : D_0 \to D_n = \mathcal{P}^{n+1}(\mathcal{L})$
Output type (from premises Γ)	Set of consequences $\text{Cl}(\Gamma) = \{\varphi \in \mathcal{L} \mid \Gamma \vdash_L \varphi\} \in H$	Family of consequence sets $\dashv_H (\Gamma) \subseteq H$	Nested family (depth n) $\delta^{(n)}(\Gamma) = \underbrace{\{\{\cdots\{\text{Cl}(\Gamma)\}\cdots\}\}}_{n \text{ nested braces}} \in D_n$
Axioms / properties (typical)	Tarskian consequence: reflexivity, monotonicity, cut, substitution invariance	Hyperreflexivity: $\Gamma \in \dashv_H (\Gamma)$; hypermonotonicity: $\Delta \in \dashv_H (\Gamma)$, $\Delta \subseteq \Theta \Rightarrow \Theta \in \dashv_H (\Gamma)$; hypercut: $\Delta \in \dashv_H (\Gamma)$, $\Theta \in \dashv_H (\Delta) \Rightarrow \Theta \in \dashv_H (\Gamma)$	Recursively inherited properties from Cl: idempotence $\delta^{(n)}(\delta^{(n)}(\Gamma)) = \delta^{(n)}(\Gamma)$; substitution invariance $\delta^{(n)}(\sigma(\Gamma)) = \sigma(\delta^{(n)}(\Gamma))$
Recovery of classical level	Base notion	If $\dashv_H (\Gamma) = \{\text{Cl}(\Gamma)\}$ (singleton-valued), then hyperlogic reduces to the original abstract logic	$n = 0$ gives $\delta^{(0)} = \text{Cl}$ (abstract logic); $n = 1$ gives $\delta^{(1)}(\Gamma) = \{\text{Cl}(\Gamma)\}$ (singleton-valued hyperlogic)
Interpretation (one sentence)	Single, deterministic closure of a premise set	Multiple admissible closures/expansions of the same premise set (nondeterministic consequence)	Multi-tiered (hierarchical) wrapping of consequence information into n nested levels (meta-inference layers)

TABLE 5. Overview of abstract propositional logic and its hyper/superhyper extensions.

- $x : E \times T \to X$ is the *state mapping*, where $x(i,t)$ is the state of component i at time t.
- $F = \big(F_i\big)_{i \in E}$ is a family of *local update functions*, where for each $i \in E$:

$$F_i : X^{N_i} \times T \longrightarrow X, \quad N_i = \{\, j \in E \mid (j,i) \in I\}$$

and $F_i\big((x(j,s))_{j \in N_i,\, s<t},\, t\big)$ determines the new state of i at time t.

These data satisfy, for all $i \in E$ and all $t \in T$ with $t > t_0$,

$$x(i,t) = F_i\big(\,(x(j,s))_{j \in N_i,\, s<t},\, t\big).$$

Definition 32 (Subsystem)**.** Let $S = (E, I, X, T, t_0, x, F)$ be a system. A *subsystem* of S is a tuple

$$S' = \big(E',\ I',\ X,\ T,\ t_0,\ x',\ F'\big)$$

where

- $E' \subseteq E$ satisfies $N_i \subseteq E'$ for every $i \in E'$.
- $I' = I \cap (E' \times E')$.
- $x' = x|_{E' \times T}$.
- $F' = (F_i)_{i \in E'}$.

Theorem 17 (Subsystem Closure)**.** *Every subsystem S' of a system S is itself a system in the sense of Definition 2.1.*

Proof: Since $E' \neq \varnothing$, $I' \subseteq E' \times E'$, T remains a totally ordered set, x' is a restriction of x, and each local update F_i for $i \in E'$ still maps $X^{N_i} \times T \to X$ with $N_i \subseteq E'$, it follows directly that S' satisfies all the conditions of a discrete-time system.

Example 16 (Two-Room Temperature Exchange System)**.** We model two adjacent rooms, A and B, exchanging heat and subject to an outside temperature variation. Take

$$E = \{A, B\}, \quad I = \{(A,A),(A,B),(B,A),(B,B)\}, \quad X = \mathbb{R}\,(\text{temperature in } ^\circ\text{C}), \quad T = \{0,1,2,\dots\}, \quad t_0 = 0.$$

The state mapping

$$x\colon E \times T \longrightarrow X, \qquad x(i,t) = \text{temperature of room } i \text{ at time } t,$$

evolves according to local update functions $F = (F_A, F_B)$ with

$$N_A = N_B = \{A, B\},$$

and

$$F_A\big((T_A,T_B),t\big) = 0.7\,T_A \;+\; 0.2\,T_B \;+\; 0.1\,T_{\text{ext}}(t),$$
$$F_B\big((T_A,T_B),t\big) = 0.3\,T_A \;+\; 0.6\,T_B \;+\; 0.1\,T_{\text{ext}}(t),$$

where the outside temperature is

$$T_{\text{ext}}(t) = 10 \;+\; 5\,\sin\!\big(\tfrac{2\pi\,t}{24}\big) \quad \big(^\circ\text{C}\big).$$

Thus for every $i \in \{A, B\}$ and $t > 0$:

$$x(i,t) = F_i\big(x(A,t-1), x(B,t-1),\, t\big).$$

Initial Conditions and First Step Let

$$x(A,0) = 20.0, \quad x(B,0) = 18.0 \quad (^\circ\text{C}).$$

Then at $t = 1$:

$$T_{\text{ext}}(1) = 10 + 5\sin\!\big(\tfrac{2\pi}{24}\big) \approx 11.29\ (^\circ\text{C}),$$
$$x(A,1) = 0.7 \cdot 20.0 + 0.2 \cdot 18.0 + 0.1 \cdot 11.29 \approx 18.73\ (^\circ\text{C}),$$
$$x(B,1) = 0.3 \cdot 20.0 + 0.6 \cdot 18.0 + 0.1 \cdot 11.29 \approx 17.93\ (^\circ\text{C}).$$

Second Step Similarly at $t = 2$:

$$T_{\text{ext}}(2) = 10 + 5\sin\!\big(\tfrac{4\pi}{24}\big) \approx 13.41,$$
$$x(A,2) = 0.7 \cdot 18.73 + 0.2 \cdot 17.93 + 0.1 \cdot 13.41 \approx 18.06,$$
$$x(B,2) = 0.3 \cdot 18.73 + 0.6 \cdot 17.93 + 0.1 \cdot 13.41 \approx 17.48.$$

Interpretation

I. Each room's new temperature is a weighted average of its own previous temperature, the neighboring room's temperature, and the outside temperature.

II. The influence graph I tells us both rooms affect each other.

III. Over time, the two rooms and the external environment drive the dynamics of the system.

Definition 33 (Discrete-Time HyperSystem)**.** Let

$$S = \big(E,\ I,\ X,\ T,\ t_0,\ x,\ F_H\big)$$

be a tuple where:

I. E is a nonempty set of *components*.

II. $I \subseteq E \times E$ is the *influence relation*.

III. X is the *state space*.

IV. T is a nonempty totally ordered set (time index) with minimum $t_0 = \min T$.

V. $x : E \times T \to X$ is the *state mapping*, so $x(i,t)$ is the state of component i at time t.

$$F_{H,i} : \mathcal{P}\big(X^{N_i}\big) \times T \;\longrightarrow\; \mathcal{P}(X), \qquad N_i = \{\, j \in E \mid (j,i) \in I\}.$$

VI. $F_H = (F_{H,i})_{i \in E}$ is a family of *local hyperupdate functions*, each

We call S a *discrete-time hypersystem* if for every $i \in E$ and $t > t_0$,

$$x(i,t) \ \in \ F_{H,i}\Big(\{\, x(j,s) \mid j \in N_i,\ s < t \},\ t \Big).$$

Remark 12 (Hyperstructure Property). Each $F_{H,i} : \mathcal{P}(X^{N_i}) \times T \to \mathcal{P}(X)$ is a *binary hyperoperation* (its codomain is a powerset), so $\big(\mathcal{P}(X^{N_i}) \times T,\ F_{H,i}\big)$ is a *hyperstructure*. Hence the entire hypersystem S carries a hyperstructure of local updates.

Theorem 18 (Recovery of Classical System). *If each hyperupdate always returns a singleton,*

$$F_{H,i}\big(A,t\big) = \{\ F_i(A,t)\ \} \quad \textit{for all } A \subseteq X^{N_i},\ t \in T,$$

then S reduces exactly to the usual discrete-time system $\big(E, I, X, T, t_0, x, (F_i)_{i\in E}\big)$.

Proof: Under the singleton assumption, the inclusion

$$x(i,t) \in F_{H,i}\big(\{x(j,s)\},t\big)$$

becomes the equality $x(i,t) = F_i\big(\{x(j,s)\},t\big)$, so each $F_{H,i}$ behaves like a classical update function F_i. All other data remain the same, recovering the standard system.

Example 17 (Two-Room Thermal HyperSystem with Uncertain Conductance). A thermal system involves components that transfer, store, or convert heat energy, typically governed by thermodynamic principles and temperature variations (cf.[33, 34, 35]). Let

$$E = \{A, B\}, \quad I = \{(A,A),(A,B),(B,A),(B,B)\}, \quad X = \mathbb{R}, \quad T = \{0,1,2,\dots\}, \quad t_0 = 0,$$

where $x(i,t)$ is the temperature (in °C) of room i at day t. Suppose the outside temperature at day t is

$$T_{\text{ext}}(t) = 10 + 5\sin\!\big(\tfrac{2\pi t}{24}\big).$$

We model uncertain heat-transfer coefficients by letting each local hyperupdate return two possible values. Concretely, for $N_A = N_B = \{A, B\}$, define

$$F_{H,A}\big(\{T_A, T_B\}, t\big) = \big\{0.6\,T_A + 0.3\,T_B + 0.1\,T_{\text{ext}}(t),\ 0.7\,T_A + 0.2\,T_B + 0.1\,T_{\text{ext}}(t)\big\},$$

$$F_{H,B}\big(\{T_A, T_B\}, t\big) = \big\{0.2\,T_A + 0.7\,T_B + 0.1\,T_{\text{ext}}(t),\ 0.3\,T_A + 0.6\,T_B + 0.1\,T_{\text{ext}}(t)\big\}.$$

Thus $S = (E, I, X, T, t_0, x, F_H)$ is a discrete-time hypersystem.

Initial condition and first update. Set

$$x(A,0) = 20.0, \quad x(B,0) = 18.0.$$

At $t = 1$,

$$T_{\text{ext}}(1) = 10 + 5\sin\!\big(\tfrac{2\pi}{24}\big) \approx 11.29.$$

Hence

$$F_{H,A}(\{20.0, 18.0\}, 1) = \{\,0.6 \cdot 20 + 0.3 \cdot 18 + 0.1 \cdot 11.29,\ 0.7 \cdot 20 + 0.2 \cdot 18 + 0.1 \cdot 11.29\} = \{18.5,\ 18.7\},$$

$$F_{H,B}(\{20.0, 18.0\}, 1) = \{\,0.2 \cdot 20 + 0.7 \cdot 18 + 0.1 \cdot 11.29,\ 0.3 \cdot 20 + 0.6 \cdot 18 + 0.1 \cdot 11.29\} = \{17.7,\ 17.9\}.$$

By the hypersystem law,

$$x(A,1) \in \{18.5, 18.7\}, \quad x(B,1) \in \{17.7, 17.9\}.$$

Thus the set of possible states at $t = 1$ is

$$\{(18.5, 17.7),\ (18.5, 17.9),\ (18.7, 17.7),\ (18.7, 17.9)\} \subseteq X^2.$$

Interpretation

I. The two values in each hyperupdate reflect uncertainty in the rooms' thermal conductance.

II. The resulting "hyperstate" at each step is a set of classical temperature vectors.

III. This hyperstructure captures nondeterministic or interval-valued evolution in a single formalism.

Definition 34 (*n*-SuperHyperSystem). Fix an integer $n \geq 1$. Define the iterated powersets

$$\mathcal{P}_0(X) = X, \quad \mathcal{P}_{k+1}(X) = \mathcal{P}\big(\mathcal{P}_k(X)\big), \quad k \geq 0.$$

An *n-superhypersystem* is a tuple

$$S^{(n)} = \big(E,\ I,\ X,\ T,\ t_0,\ x,\ F^{(n)}\big),$$

where

$$F^{(n)} = \big(F_i^{(n)}\big)_{i \in E}, \quad F_i^{(n)} : \mathcal{P}_n\big(X^{N_i}\big) \times T \longrightarrow \mathcal{P}_n(X),$$

and the evolution law is, for $t > t_0$,

$$x(i,t) \ \in \ \big[F_i^{(n)}\big(\{\, x(j,s)\},t\big)\big]_{(n)},$$

where $\{\cdot\}_{(n)}$ indicates selection down through the n–fold powerset.

Theorem 19 (n-SuperHyperstructure Property)**.** *Each* $F_i^{(n)} \colon \mathcal{P}_n(X^{N_i}) \times T \ \to \ \mathcal{P}_n(X)$ *is a binary* $(2,n)$*-superhyperoperation. Therefore* $\big(\mathcal{P}_n(X^{N_i}) \times T,\ F_i^{(n)}\big)$ *is a* $(2,n)$*-superhyperstructure of local updates.*

Proof: By construction, the codomain of each $F_i^{(n)}$ is $\mathcal{P}_n(X)$, the n–th iterated powerset of X. This exactly matches the definition of an (m,n)-superhyperoperation (here $m = 2$ arguments), so the collection $F^{(n)}$ endows $S^{(n)}$ with a superhyperstructure.

Example 18 (Two-Room 2-SuperHyperSystem: Calibration and Model Uncertainty)**.** Consider again two adjacent rooms A, B with state-space $X = \mathbb{R}$ (temperature in °C), time-index $T = \{0, 1, 2, \dots\}$, initial time $t_0 = 0$, and influence graph $I = \{(A, A), (A, B), (B, A), (B, B)\}$. We now model two levels of uncertainty:

Level 1 (Hyperupdate): For each room $i \in \{A, B\}$, the local hyperupdate returns a set of two candidate temperatures reflecting measurement noise:

$$F_{H,i}\big(\{T_A, T_B\}, t\big) = \big\{\ f_i^{(1)}(T_A, T_B, t),\ f_i^{(2)}(T_A, T_B, t)\big\},$$

where—for example—at $t = 1$ with $T_A = 20.0$, $T_B = 18.0$ and outside $T_{\text{ext}}(1) \approx 11.29$:

$$f_A^{(1)} = 0.6\,T_A + 0.3\,T_B + 0.1\,T_{\text{ext}} = 18.5, \quad f_A^{(2)} = 0.7\,T_A + 0.2\,T_B + 0.1\,T_{\text{ext}} = 18.7,$$
$$F_{H,A}(\{20, 18\}, 1) = \{18.5,\ 18.7\}, \quad F_{H,B}(\{20, 18\}, 1) = \{17.7,\ 17.9\}.$$

Level 2 (SuperHyperupdate): We further admit two distinct calibration models, each yielding its own Level 1 hyperupdate. Thus

$$F_i^{(2)}\big(\{T_A, T_B\}, t\big) = \big\{F_{H,i}^{(\text{cal1})}(\{T_A, T_B\}, t),\ F_{H,i}^{(\text{cal2})}(\{T_A, T_B\}, t)\big\},$$

where each $F_{H,i}^{(\text{cal k})}$ is itself a Level 1 hyperupdate set. Concretely at $t = 1$:

$$F_A^{(2)}(\{20, 18\}, 1) = \big\{\{18.5, 18.7\},\ \{19.0, 18.8\}\big\}, \quad F_B^{(2)}(\{20, 18\}, 1) = \big\{\{17.7, 17.9\},\ \{18.0, 17.8\}\big\}.$$

Resulting 2-SuperHyperstate at $t = 1$. The collection of all possible Level 1 hyperstates under both calibration models is

$$\Big\{\ H_A \times H_B \ \Big|\ H_A \in F_A^{(2)}(\{20, 18\}, 1),\ H_B \in F_B^{(2)}(\{20, 18\}, 1)\Big\},$$

i.e. the four sets

$$\{18.5, 18.7\} \times \{17.7, 17.9\}, \quad \{18.5, 18.7\} \times \{18.0, 17.8\},$$
$$\{19.0, 18.8\} \times \{17.7, 17.9\}, \quad \{19.0, 18.8\} \times \{18.0, 17.8\}.$$

Each of these is itself a set of two-room temperature pairs, giving a *second-level* hyperstructure.

Interpretation.

I. Level 1 captures sensor/measurement noise (two possible readings per room).

II. Level 2 captures calibration model uncertainty (two distinct hyperupdates).

III. The 2-superhyperstate at each time step is a set of hyperstates, each hyperstate being a set of possible temperature pairs.

IV. A nested selection (choose one calibration model, then one reading per room) recovers a classical temperature trajectory.

Example 19 (Discrete-Time Epidemiological SIR Model)**.** The SIR model is an epidemiological model dividing a population into susceptible, infected, and recovered compartments to study disease spread (cf.[36, 37, 38]). We model the spread of an infectious disease in a fixed population of size N. Let

$$E = \{S, I, R\}, \qquad X = [0, N] \subset \mathbb{R}, \qquad T = \{0, 1, 2, \dots\}, \qquad t_0 = 0.$$

Here $x(S,t)$, $x(I,t)$, $x(R,t)$ denote the numbers of Susceptible, Infectious, and Recovered individuals at day t. Define the influence relation

$$I = \{(S,S), (S,I), (I,S), (I,I), (I,R), (R,R)\},$$

so that

$$N_S = \{S, I\}, \quad N_I = \{S, I\}, \quad N_R = \{I, R\}.$$

Choose infection rate $\beta > 0$ and recovery rate $\gamma > 0$. The local update functions $F = (F_S, F_I, F_R)$ are

$$\begin{aligned} F_S\big((S_t, I_t), t\big) &= S_t \; - \; \beta \, \frac{S_t \, I_t}{N}, \\ F_I\big((S_t, I_t), t\big) &= I_t \; + \; \beta \, \frac{S_t \, I_t}{N} \; - \; \gamma \, I_t, \\ F_R\big((R_t, I_t), t\big) &= R_t \; + \; \gamma \, I_t. \end{aligned}$$

Thus for each $i \in \{S, I, R\}$ and $t > 0$,

$$x(i,t) \; = \; F_i\big((x(j, t-1))_{j \in N_i}, \, t-1 \big).$$

Parameters and Initial State Let $N = 1000$, $\beta = 0.3/N$, $\gamma = 0.1$, and

$$x(S,0) = 999, \quad x(I,0) = 1, \quad x(R,0) = 0.$$

First Day ($t = 1$) Compute new infections and recoveries:

$$\beta \, \frac{999 \cdot 1}{1000} \approx 0.2997, \quad \gamma \cdot 1 = 0.1.$$

Hence

$$\begin{aligned} x(S,1) &= 999 - 0.2997 \approx 998.7003, \\ x(I,1) &= 1 + 0.2997 - 0.1 \approx 1.1997, \\ x(R,1) &= 0 + 0.1 = 0.1. \end{aligned}$$

Second Day ($t = 2$) With $S_1 \approx 998.7003$, $I_1 \approx 1.1997$:

$$\beta \, \frac{998.7003 \cdot 1.1997}{1000} \approx 0.3595, \quad \gamma \cdot 1.1997 \approx 0.11997.$$

Thus

$$\begin{aligned} x(S,2) &\approx 998.7003 - 0.3595 = 998.3408, \\ x(I,2) &\approx 1.1997 + 0.3595 - 0.11997 = 1.4392, \\ x(R,2) &\approx 0.1 + 0.11997 = 0.21997. \end{aligned}$$

Interpretation

I. S_t decreases by the number of new infections $\beta S_t I_t/N$.

II. I_t increases by new infections and decreases by recoveries γI_t.

III. R_t increases by γI_t.

IV. This discrete-time system captures "memoryless" transitions in an epidemic, suitable for daily-step simulations.

For reference, an overview of discrete-time systems and their hyper/superhyper extensions is provided in Table 6.

Aspect	**Discrete-Time System**	**Discrete-Time HyperSystem**	**Discrete-Time n-SuperHyperSystem**
Base object	A tuple $S = (E, I, X, T, t_0, x, (F_i)_{i\in E})$ with deterministic local updates.	A tuple $S_H = (E, I, X, T, t_0, x, (F_{H,i})_{i\in E})$ with set-valued local updates.	A tuple $S^{(n)} = (E, I, X, T, t_0, x, (F_i^{(n)})_{i\in E})$ with n-fold set-valued local updates.
Time index	Discrete $T = \{0, 1, 2, \dots\}$ (or any totally ordered set with minimum t_0).	Same T as classical.	Same T as classical.
Influence structure	$I \subseteq E \times E$, with neighborhood $N_i = \{j \in E \mid (j, i) \in I\}$.	Same I and N_i.	Same I and N_i.
State space	X (e.g., $\mathbb{R}$, $[0, N]$, finite alphabets).	X (underlying crisp state space remains unchanged).	X (underlying crisp state space remains unchanged).
Local update type	$F_i : X^{N_i} \times T \to X$.	$F_{H,i} : \mathcal{P}(X^{N_i}) \times T \to \mathcal{P}(X)$.	$F_i^{(n)} : \mathcal{P}_n(X^{N_i}) \times T \to \mathcal{P}_n(X)$, where $\mathcal{P}_0(Y) = Y$ and $\mathcal{P}_{k+1}(Y) = \mathcal{P}(\mathcal{P}_k(Y))$.
Evolution law (one step)	For $t > t_0$, $x(i,t) = F_i\big((x(j,s))_{j\in N_i,\, s<t},\, t\big)$ (or in Markov form $x(i,t) = F_i((x(j,t-1))_{j\in N_i}, t)$).	For $t > t_0$, $x(i,t) \in F_{H,i}\Big(\{x(j,s) \mid j \in N_i,\, s < t\},\, t\Big)$.	For $t > t_0$, $x(i,t)$ is obtained by nested selection from the n-level output $F_i^{(n)}(\cdot, t) \in \mathcal{P}_n(X)$ (choose down through n powerset levels).
Uncertainty meaning	No intrinsic uncertainty: one next state per component at each time.	Nondeterministic / multi-valued update: several admissible next states at each time.	Hierarchical uncertainty: sets of sets of $\cdots$ of states (up to n levels), e.g., uncertainty about models, then about parameters, then about measurements.
Recovery of classical case	— (baseline).	If $F_{H,i}(A, t) = \{F_i(A, t)\}$ (singleton output) for all i, then the hypersystem collapses to the classical system.	If each $F_i^{(n)}$ is built as n nested singleton-lifts of a deterministic F_i, then $S^{(n)}$ collapses to the classical system after unwrapping.
Typical examples	Discrete SIR, discrete thermal exchange, finite-state automata.	Thermal exchange with uncertain conductance (two candidate updates), nondeterministic control policies.	Two-level calibration + sensor-noise (a 2-superhypersystem), multi-tier model ensembles with nested scenario sets.

TABLE 6. Overview of discrete-time systems and their hyper/superhyper extensions. For a set Y, $\mathcal{P}(Y)$ denotes the powerset of Y, and $\mathcal{P}_n(Y)$ denotes the n-fold iterated powerset defined by $\mathcal{P}_0(Y) = Y$ and $\mathcal{P}_{k+1}(Y) = \mathcal{P}(\mathcal{P}_k(Y))$.

3|Conclusion and Future Works

In this paper, we explored the extension of Queues, Markov Chains, Intervals, Logic, and Systems into hyperstructural and superhyperstructural frameworks, examining their underlying mathematical properties and providing detailed illustrative examples.

In future work, we aim to further enhance the modeling of real-world phenomena by extending these frameworks using various uncertainty-handling structures, including Fuzzy Sets [39], Intuitionistic Fuzzy Sets [40], Hesitant Fuzzy Sets [41], Neutrosophic Sets [42], Picture Fuzzy Sets [43], Hyperfuzzy Sets [44], Vague Sets [45], QuadriPartitioned Neutrosophic Sets [46], Uncertain Sets [47, 48], Hyperneutrosophic Sets [49], and Plithogenic Sets [50]. We hope that these extended frameworks will facilitate further research into applications such as decision-making and queueing models.

Funding

No external funding or financial support was provided for this study.

Acknowledgments

The authors wish to thank all colleagues and mentors whose feedback and encouragement enriched this work. We are grateful to the community of researchers whose foundational contributions informed our developments. Special appreciation goes to the institutions that offered resources and technical infrastructure throughout this project.

Data Availability

This manuscript presents purely conceptual work without empirical data. Scholars interested in these ideas are invited to undertake experimental or case-study research to substantiate and extend the proposed frameworks.

Ethical Approval

This paper involves no human or animal subjects and thus did not require ethics committee review or approval.

Use of Generative AI and AI-Assisted Tools

We use generative AI and AI-assisted tools for tasks such as English grammar checking, and We do not employ them in any way that violates ethical standards.

Conflicts of Interest

The authors declare that there are no competing interests concerning the content or publication of this article.

Disclaimer

The theoretical models and propositions herein have not yet been subjected to practical validation. Readers should independently verify all citations and be aware that inadvertent inaccuracies may remain. The opinions expressed are those of the authors and do not necessarily represent the views of affiliated organizations.

References

[1] Smarandache, F. (2020). Extension of Hypergraph to n-Superhypergraph and to Plithogenic n-Superhypergraph, and extension of Hyperalgebra to n-Ary (classical-/neutro-/anti-) Hyperalgebra. *Neutrosophic sets and systems, 33*, 290–296. https://doi.org/10.5281/zenodo.3783103

[2] Smarandache, F. (2017). Hyperuncertain, superuncertain, and superhyperuncertain sets/logics/probabilities/statistics. *Critical review: A journal of politics and society, 14*, 10–19. https://digitalrepository.unm.edu/math_fsp/788/

[3] Smarandache, F. (2024). Foundation of superhyperstructure & Neutrosophic superhyperstructure. *Neutrosophic sets and systems*, *63*(2024), 367-381. https://fs.unm.edu/nss8/index.php/111/article/view/3896

[4] Hjorth, G. (2005). T. Jech. Set theory. The third millennium edition, revised and expanded. Springer-Verlag, Berlin, 2003, viii+ 769 pp. *Bulletin of symbolic logic*, *11*(2), 243-245. https://doi.org/10.1017/S1079898600003358

[5] Vougioukli, S. (2020). Helix hyperoperation in teaching research. *Science & philosophy*, *8*(2), 157-163. http://dx.doi.org/10.23756/sp.v8i2.557
[6] Adams, R. (2012). Active queue management: A survey. *IEEE communications surveys & tutorials*, *15*(3), 1425-1476. https://doi.org/10.1109/SURV.2012.082212.00018
[7] Kunniyur, S., & Srikant, R. (2001). Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. *ACM SIGCOMM computer communication review*, *31*(4), 123-134. https://doi.org/10.1145/964723.383069
[8] Bose, S. K. (2013). *An introduction to queueing systems*. Springer Science & Business Media. https://doi.org/10.1007/978-1-4615-0001-8
[9] Xu, S. H., Gao, L., & Ou, J. (2007). Service performance analysis and improvement for a ticket queue with balking customers. *Management science*, *53*(6), 971-990. https://doi.org/10.1287/mnsc.1060.0660
[10] Fazli Alias, M. F. (2007). *Front desk customer service for queue management system* [Thesis]. https://www.semanticscholar.org/paper/Front-desk-customer-service-for-queue-managementFa-zli/191b6f8b4009e21ee466cfed7c879b1a7a9835c6
[11] Hanukov, G., Anily, S., & Yechiali, U. (2020). Ticket queues with regular and strategic customers. *Queueing systems*, *95*(1), 145-171. https://doi.org/10.1007/s11134-020-09647-x
[12] Yin, G. G., & Zhang, Q. (2005). *Discrete-time Markov chains: Two-time-scale methods and applications*. New York, NY: Springer New York. https://www.amazon.com/Discrete-Time-Markov-Chains-Two-Time-Scale-Applications/dp/038721948X
[13] Craig, B. A., & Sendi, P. P. (2002). Estimation of the transition matrix of a discrete-time Markov chain. *Health economics*, *11*(1), 33-42. https://doi.org/10.1002/hec.654
[14] Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the royal statistical society series b: Statistical methodology*, *72*(3), 269-342. https://doi.org/10.1111/j.1467-9868.2009.00736.x
[15] Karush, J. (1961). On the Chapman-Kolmogorov equation. *The annals of mathematical statistics*, *32*(4), 1333-1337. https://www.jstor.org/stable/2237931
[16] Bauer, P., Thorpe, A., & Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, *525*(7567), 47-55. https://doi.org/10.1038/nature14956
[17] Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. *Quarterly journal of the royal meteorological society*, *112*(474), 1177-1194. https://doi.org/10.1002/qj.49711247414
[18] Lynch, P. (2008). The origins of computer weather prediction and climate modeling. *Journal of computational physics*, *227*(7), 3431-3444. https://doi.org/10.1016/j.jcp.2007.02.034
[19] Riordan, D., & Hansen, B. K. (2002). A fuzzy case-based system for weather prediction. *Engineering intelligent systems for electrical engineering and communications*, *10*(3), 139-146. https://www.chebucto.ns.ca/Science/AIMET/cs/riordan_and_hansen_2002.pdf
[20] Salawu, E. Y., Awoyemi, O. O., Akerekan, O. E., Afolalu, S. A., Kayode, J. F., Ongbali, S. O., ... & Edun, B. M. (2023). Impact of maintenance on machine reliability: A review. *E3S web of conferences* (Vol. 430, p. 01226). EDP Sciences. https://doi.org/10.1051/e3sconf/202343001226
[21] Kokieva, G. E., Voinash, S. A., Maksimovich, K. Y., Sokolova, V. A., Ivanov, A. A., & Panov, A. Y. (2020). On calculation and assessment of machine reliability. *Journal of physics: Conference series* (Vol. 1679, No. 4, p. 042029). IOP Publishing. https://doi.org/10.1088/1742-6596/1679/4/042029
[22] Das, K., Lashkari, R. S., & Sengupta, S. (2007). Machine reliability and preventive maintenance planning for cellular manufacturing systems. *European journal of operational research*, *183*(1), 162-180. https://doi.org/10.1016/j.ejor.2006.09.079
[23] Gillman, L. (1952). On intervals of ordered sets. *Annals of mathematics*, *56*(3), 440-459. https://doi.org/10.2307/1969653
[24] Schröder, B. (2016). *Ordered sets*. Springer International Publishing. https://doi.org/10.1007/978-3-319-29788-0

[25] Sengönül, M., & Eryılmaz, A. (2010). On the sequence spaces of interval numbers. *Thai journal of mathematics*, *8*(3), 503-510. https://d1wqtxts1xzle7.cloudfront.net/77724438/317-libre.pdf

[26] Cai, H., Zhao, W., & Njock, P. G. A. (2024). Multi-tier scheduling algorithm of dispatching systems for urban water logging. *Smart construction and sustainable cities*, *2*(1), 3. https://doi.org/10.1007/s44268-024-00027-5

[27] Alnaser, A. A. M. A. A., Saloum, S. S., Sharadqh, A. A., & Hatamleh, H. (2024). Optimizing multi-tier scheduling and secure routing in edge-assisted software-defined wireless sensor network environment using moving target defense and AI techniques. *Future internet*, *16*(11), 386. https://doi.org/10.3390/fi16110386

[28] Buvac, S., & Mason, I. A. (1993). Propositional logic of context. *Proceedings of the eleventh national conference on artificial intelligence (AAAI-93)* (pp. 412–419). AAAI Press. https://cdn.aaai.org/AAAI/1993/AAAI93-062.pdf

[29] Büning, H. K., & Lettmann, T. (1999). *Propositional logic: Deduction and algorithms*. Cambridge University Press. https://scispace.com/pdf/propositional-logic-deduction-and-algorithms-3lvvbc0yqg.pdf

[30] Aoki, M. (2016). *Optimization of stochastic systems: Topics in discrete-time systems*. Elsevier. https://books.google.com/books?id=T3RnDAAAQBAJ&printsec=frontcover#v=onepage&q&f=false

[31] Helton, J. W. (1974). Discrete time systems, operator models, and scattering theory. *Journal of functional analysis*, *16*(1), 15-38. https://doi.org/10.1016/0022-1236(74)90069-X

[32] Amato, F., & Ariola, M. (2005). Finite-time control of discrete-time linear systems. *IEEE transactions on automatic control*, *50*(5), 724-729. https://doi.org/10.1109/TAC.2005.847042

[33] Patel, V. K., Savsani, V. J., & Tawhid, M. A. (2019). *Thermal system optimization*. Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-030-10477-1

[34] Kim, S. M., Oh, S. D., Kwon, Y. H., & Kwak, H. Y. (1998). Exergoeconomic analysis of thermal systems. *Energy*, *23*(5), 393-406. https://doi.org/10.1016/S0360-5442(97)00096-0

[35] Huide, F., Xuxin, Z., Lei, M., Tao, Z., Qixing, W., & Hongyuan, S. (2017). A comparative study on three types of solar utilization technologies for buildings: Photovoltaic, solar thermal and hybrid photovoltaic/thermal systems. *Energy conversion and management*, *140*, 1-13. https://doi.org/10.1016/j.enconman.2017.02.059

[36] Rodrigues, H. S. (2016). *Application of SIR epidemiological model: New trends*. https://doi.org/10.48550/arXiv.1611.02565

[37] Satsuma, J., Willox, R., Ramani, A., Grammaticos, B., & Carstea, A. S. (2004). Extending the SIR epidemic model. *Physica A: Statistical mechanics and its applications*, *336*(3-4), 369-375. https://doi.org/10.1016/j.physa.2003.12.035

[38] Yusuf, T. T., & Benyah, F. (2012). Optimal control of vaccination and treatment for an SIR epidemiological model. *World journal of modelling and simulation*, *8*(3), 194-204. https://www.wjms.org.uk/journal/1746-7233WJMS/wjmsvol08no03paper04.pdf

[39] Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, *8*(3), 338-353. https://doi.org/10.1016/S0019-9958(65)90241-X

[40] Atanassov, K. T. (2020). Circular intuitionistic fuzzy sets. *Journal of intelligent & fuzzy systems*, *39*(5), 5981-5986. https://doi.org/10.3233/JIFS-189072

[41] Torra, V. (2010). Hesitant fuzzy sets. *International journal of intelligent systems*, *25*(6), 529-539. https://doi.org/10.1002/int.20418

[42] Smarandache, F. (1998). *Neutrosophy: Neutrosophic probability, set, and logic: Analytic synthesis & synthetic analysis*. American Research Press. https://philpapers.org/rec/Smannp

[43] Hatamleh, R., Al-Husban, A., Zubair, S. A. M., Elamin, M., Saeed, M. M., Abdolmaleki, E., ... & Khattak, A. M. (2025). Ai-assisted wearable devices for promoting human health and strength using

[44] Jun, Y. B., Hur, K., & Lee, K. J. (2017). Hyperfuzzy subalgebras of Bck/Bci-algebras. *Annals of fuzzy mathematics and informatics*, 2017-11. https://www.internetdownloadmanager.com/register/new_faq/chrome_extension2.html

[45] Lu, A., & Ng, W. (2005). Vague sets or intuitionistic fuzzy sets for handling vague data: Which one is better? *International conference on conceptual modeling* (pp. 401-416). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11568322_26

[46] Saeed, M. M., Hatamleh, R., AbdEl-latif, A. M., Al-Husban, A. A. H., Attaalfadeel, H. M., Fujita, T., ... & Khattak, A. M. (2025). A breakthrough approach to quadri-partitioned Neutrosophic soft topological spaces. *European journal of pure and applied mathematics*, *18*(2), 5845-5845. https://doi.org/10.29020/nybg.ejpam.v18i2.5845

[47] Fujita, T., & Smarandache, F. (2025). A unified framework for U-structures and functorial structure: Managing super, hyper, superhyper, tree, and forest uncertain over/under/off models. *Neutrosophic sets and systems, 91*, 337–378. https://doi.org/10.5281/zenodo.16754933

[48] Fujita, T., & Smarandache, F. (2025). *A dynamic survey of fuzzy, intuitionistic fuzzy, Neutrosophic, plithogenic, and extensional sets*. Infinite Study. https://books.google.com/books?id=dGahEQAAQBAJ&printsec=frontcover#v=onepage&q&f=false

[49] Settu, K., & Jayalakshmi, M. (2025). Prediction of risk factor in hepatitis diagnosis using interval value Hyperneutrosophic and Einstein aggregated operations with extended MCDM. *Expert systems with applications, 302*, 130364. https://doi.org/10.1016/j.eswa.2025.130364

[50] Smarandache, F. (2018). Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and Neutrosophic sets – revisited. *Neutrosophic sets and systems, 21*, 153–166. https://doi.org/10.5281/zenodo.1408740